



**Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE GRAU

TÍTOL: SPK-Connect

AUTORS: Souto Hernandez, Daniel

DATA DE PRESENTACIÓ: 30-5-2018

COGNOMS: Souto Hernández

NOM: Daniel

TITULACIÓ: Enginyeria electrònica industrial i automàtica

PLA: 2013/2014

DIRECTOR: Luis Miguel Muñoz

DEPARTAMENT: IIEC

QUALIFICACIÓ DEL TFG

<u>TRIBUNAL</u>		
PRESIDENT	SECRETARI	VOCAL



--

DATA DE LECTURA:30-5-18

--

AQUEST PROJECTE TE EN COMPTE ASPECTES MEDIAMBIENTALS: ☐

SI ☐ NO

.....

RESUMEN

El objetivo de este trabajo de fin de carrera es la realización de un prototipo que permita dar información complementaria sobre el estado de una máquina de enfardado industrial.

Junto con la empresa CONTROLPACK.SL, líder en el sector del “packaging”, se pretende realizar el proyecto SPK-Connect, un proyecto basado en la industria 4.0 que pretende mejorar la eficiencia de la empresa y a la vez satisfacer al cliente.

En el siguiente trabajo hablaremos de la empresa CONTROLPACK.SL para poder entender el propósito del SPK-Connect. De la misma manera, veremos los antecedentes y como se pretende abordar el problema que se nos plantea.

Veremos cómo podemos integrar conceptos tan innovadores en la industria como es el “Internet de las cosas” y la nombrada “Industria 4.0”, todo ello con la finalidad de hacer una industria basada en el Lean Management.

Para la creación del primer prototipo del SPK-Connect, realizaremos una pequeña simulación de un ciclo de enfardado de una máquina automática de enfardado de palets.

Para hacer dicho prototipo, trabajaremos con el ya conocido Arduino, una plataforma que nos permitirá ejecutar las instrucciones que necesitemos, así como todos los sensores y componentes que requerimos.

Este trabajo de fin de grado pretende ser la chispa que se necesita para poder llevar acabo a nivel industrial el SPK-Connect.

Paraules clau (màxim 10):

ControlPack	Industria 4.0	Internet de las cosas	Lean Management
Arduino	Packaging		

TABLA DE CONTENIDO

Capítulo I.	INTRADUCCIÓN.....	12
1.1	Contexto y motivaciones del proyecto.....	12
1.2	Planteamiento y objetivos del proyecto	13
1.4	Justificaciones	16
1.5	Antecedentes, el R-connect.....	19
1.6	Sumario contenido del proyecto	20
1.7	Planificación	21
Capítulo II.	ESTADO DEL ARTE	22
2.1	Industria 4.0	22
2.1.1	El Internet de las cosas	24
1.3	GSM.....	26
2.3	GPRS	28
2.4	Comandos AT.....	30
2.5	RFID	33
2.6	Plataforma Arduino.....	35
2.6.1	Entorno de desarrollo.....	37
2.6.2	Shield GSM/GPRS	39
2.6.3	Shield RFID.....	41
Capítulo III.	DESARROLLO DEL PROTOTIPO	42
3.1	Objetivos del prototipo	42
3.2	Diseción del problema.	43
3.2.1	Lectura del Tag.....	44

3.2.2	Control del Arduino por SMS	45
3.3	Montaje	46
3.3.1	Material.....	46
3.3.2	Presupuesto.....	50
3.3.2	Conexionado.....	51
3.4	Principio de funcionamiento	53
3.5	Códigos	57
3.6	Simulación	64
Capítulo IV.	LA WRAPSTER.....	72
4.1	Listado de alarmas de la WRAPSTER	74
4.2	Diagrama de secuencia de la WRAPSTER	77
4.3	Comunicaciones Modem 2G GPRS GSM.....	80
4.4	Gestión de datos.....	81
4.5	Trabajos pendientes	83
4.5.1	Plataforma SMS.....	83
4.5.2	Carro portabobinas	86
Conclusiones.....		90
Bibliografía.....		¡Error! Marcador no definido.
Anexo I	CONTROLPACK	95
	Carta de presentación	95
	Servicio tecnico	96
	Maquinaria	97
	Envolvedora de palets	97
	Envolvedoras horizontales	98

Retractiladoras	100
Anexo II CODIGO arduino	101

Índice de figuras

Ilustración 1 Universo Arduino	16
Ilustración 2 Sistema de telecomunicaciones	17
Ilustración 3 Sistema RFID en empresa	18
Ilustración 4 R-Connect.....	19
Ilustración 5 Revoluciones Industriales.....	23
Ilustración 6 Concepto del “Internet de las cosas”	24
Ilustración 7 Expectativas de crecimiento de dispositivos inteligentes interconectados	25
Ilustración 8 Arquitectura de red del sistema GSM	27
Ilustración 9 Arquitectura de red del sistema GPRS	29
Ilustración 10 Sistema de Radio Frecuencia	33
Ilustración 11 Arduino UNO	35
Ilustración 12 IDE del Arduino	37
Ilustración 13 Monitor Serie.....	38
Ilustración 14 Shields SIM 900 GPRS/GSM	39
Ilustración 15 Colocación de los jumpers.....	39
Ilustración 16 Conexión de SIM900 con Arduino	40
Ilustración 17 Encendido de los LEDs en la Shield SIM 900	40
Ilustración 18 Shield RFID	41
Ilustración 19 Conexión de Shield RFID y Arduino	41
Ilustración 20 Módulos del prototipo	43

Ilustración 21 Diagrama de flujo RFID	44
Ilustración 22 Diagrama de flujo SIM900	45
Ilustración 23 Tabla de precios.....	50
Ilustración 24 Tabla de conexiones	51
Ilustración 25 Esquema de conexiones.....	52
Ilustración 26 Peso del film Ilustración 27 Panel de programación de una enfardadora	54
Ilustración 28 Diagrama de flujo del prototipo	56
Ilustración 29 Envolvedora de palets ROTOTECH 3000	64
Ilustración 30 Captura pantalla monitor Serial.....	64
Ilustración 31 Captura pantalla monitor Serial.....	65
Ilustración 32 Encendido LED blanco al pulsar botón verde	65
Ilustración 33 Encendido LED rojo al pulsar botón verde y no detectar tag	
Ilustración 34 Captura pantalla monitor Serial.....	66
Ilustración 35 Consumo de film	67
Ilustración 36 Activación de alarmas	67
Ilustración 37 Localización de la máquina	68
Ilustración 38 Ciclos de la máquina	68
Ilustración 39 Alerta film gastado	69
Ilustración 40 Captura pantalla monitor Serial.....	69
Ilustración 41 SMS Máquina.....	69
Ilustración 42 LED azul encendido al recibir una petición	70
Ilustración 43 SMS Máquina.....	70
Ilustración 44 LED rojo al pulsarse botón alarma	Ilustración 45 SMS
Máquina	71

Ilustración 46 Technoplat	Ilustración 47 Flejadora 2200	73
Ilustración 48 WRAPSTER.....		73
Ilustración 49 Tabla de alarmas de la WRAPSTER y entrada de datos.....		74
Ilustración 50 Panel de control de la WRAPSTER		75
Ilustración 51 Sistema para contar cambios de bobina.....		75
Ilustración 52 Datos de la WRAPSTER.....		76
Ilustración 53 Panel para definir programa		79
Ilustración 54 Modem MTX-65i		80
Ilustración 55 Página principal del servidor de Reisopack.....		81
Ilustración 56 Histórico de producción		82
Ilustración 57 Aprovisionamiento de consumibles		82
Ilustración 58 Histórico de alarmas de la máquina.....		82
Ilustración 59 diagrama del sistema		83
Ilustración 60 Ejemplo SMS		84
Ilustración 61 Layout carro bobina.....		86
Ilustración 62 Tag de la impresora		87
Ilustración 63 Antena RFID STIX	Ilustración 64 Lector RFID portátil	
WORKABOUT PRO G4		88
Ilustración 65 Célula de carga		89
Ilustración 66 Delegaciones y servicio técnico ControlPack		96
Ilustración 67 Robot S6	Technoplat 3000	97
Ilustración 68 Orbital	Spiror	98
Ilustración 69 Robotape CF	SUPERBOX	99
Ilustración 70 SPK Campana	Túnel de retactilado	100

CAPITULO I. INTRODUCCIÓN

1.1 CONTEXTO Y MOTIVACIONES DEL PROYECTO

Este proyecto se realiza como Trabajo de Fin de Grado(**TFG**) de la carrera de Ingeniería en electrónica industrial y automática, de la Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú (**EPSEVG**) de la Universidad Politècnica de Catalunya (**UPC**). Es un proyecto de modalidad B realizado en la empresa **Control Pack Systems S.L.** Para más información sobre la actividad de la empresa podéis consultar su página web (**CONTROLPACK**, 2018).

Durante el verano del 2017, me incorpore en la empresa Control Pack System (Anexo I) como desarrollador de proyectos, encargado de proseguir con un proyecto de innovación aparte de ser el responsable de una de las secciones del departamento de maquinaria, precintadoras y formadoras de cajas.

A medida que el verano acababa y llegaba el momento de volver a la universidad, planteé la idea de poder hacer las prácticas de empresa con ellos, no hubo ningún problema y me dieron todas las facilidades para poder seguir trabajando a distancia e incluso de poder hacer el trabajo de fin de grado.

Propuse una idea a la empresa, una idea de I+D que se podría aplicar a sus máquinas y a todo su sistema. El SPK-Connect, pretende ser una plataforma que permita de manera remota conocer el estado de una de las máquinas de enfardado, dando datos a la empresa y a los técnicos.

El 70% de la facturación de Control Pack, proviene de la venta de film industrial, es decir el consumible de las máquinas enfardadoras. Este proyecto pretende ser capaz de calcular y transmitir el consumo de film que puede tener una máquina, con lo cual Control Pack puede adaptar su producción a la demanda.

El proyecto del SPK-Connect partía de la base de otro sistema de teleasistencia de la casa ROBOPAC, el R-Connect, seguidamente daré más información sobre este sistema. En su página web podéis encontrar una presentación de este sistema (ROBOPAC R-Connect, 2018).

Crear un sistema basado en el R-Connect, capaz de transmitir información de la máquina a un servidor web. Identificar si la bobina de film que está utilizando la máquina es de Control Pack y hacer una función para saber si la bobina de film está a punto de acabarse.

A finales de octubre solicite una reunión con las diferentes partes implicadas para poder definir objetivos y trazar un camino que seguir, repartir el trabajo, etc. Actualmente soy el encargado del proyecto y mi misión es poderlo llevar a cabo el SPK-Connect junto con un equipo de técnicos y la colaboración con otras empresas.

En un principio parecía un buen trabajo de fin de grado, un proyecto en el que entraba: búsqueda de información, autoaprendizaje, elaboración de un prototipo, gestión con la empresa y trabajo colaborativo con otra empresa.

1.2 PLANTEAMIENTO Y OBJETIVOS DEL PROYECTO

Como se introducía en el apartado anterior, la idea de partida para el desarrollo del proyecto se basa en el diseño de una plataforma que envíe información al cliente o al técnico sobre el estado de la máquina y su consumible. Mediante un servidor web poder mantener un historial del funcionamiento de la máquina y el uso que da de ella el cliente.

El tipo de tecnología que utilizaremos para realizar el prototipo será siempre la más parecida a la tecnología que incluirá en un futuro el proyecto final.

Una de las premisas que fijamos en un principio fue reducir al máximo los costes de fabricación del dispositivo, de manera que obtuviésemos como resultado una herramienta de bajo coste que estuviese al alcance de cualquier cliente.

Tras evaluar los costes que conllevaría la elección de cada una de las tecnologías planteadas y la posibilidad de que los clientes y técnicos tuviesen a su disposición los

medios necesarios para contar con esas tecnologías, siempre cobró más fuerza la idea de utilizar la shield GPRS/GSM como sistema de comunicación. Decantarnos por el uso del Wi-Fi o Ethernet no era viable a causa del destino de estas máquinas al tratar de una fábrica donde no se puede o no se aconseja una instalación de Ethernet o no hay señal de Wi-Fi.

Por otro lado, otro de los planteamientos que tenían más peso era utilizar la tecnología RFID como sistema de identificación de bobinas, a corto plazo esta tecnología nos serviría para solucionar uno de los problemas que tenía la empresa y a largo plazo, esta tecnología podría expandirse a toda la logística de la empresa.

Control Pack tiene un sistema de financiación de máquinas envolvedoras llamado Pay Per Use. En resumen, se hace un estudio del uso del film que actualmente tiene el cliente, seguidamente se le hace una oferta que consiste en dejar en concesión la máquina al cliente con la condición de que compre las bobinas de film a Control Pack con un incremento. El contrato puede ser de 3 o 5 años dependiendo del consumo y el importe de la máquina. Una vez finalizado el contrato pueden pasar varios casos. Por ejemplo, el cliente no ha llegado a la cuota de film necesaria o finaliza el contrato y no ha acabado de pagar la máquina... En definitiva, este proyecto también quiere ser un sistema para blindar los contratos el Pay Per Use, haciendo que la máquina trabaje exclusivamente con bobinas de Control Pack y facilitar un registro del consumo del cliente.

A finales de diciembre de 2017 durante el congreso de ventas que celebraba Control Pack, tuve la suerte y el placer de conocer a uno de los ingenieros detrás del R-Connect, mantuve una reunión con el dónde le planteé los problemas que teníamos, como queríamos solucionarlos etc. Tras finalizar la reunión, el SPK-Connect cambiaría profundamente.

El proyecto se puede dividir en cuatro grandes apartados:

1. Planteamiento, búsqueda de información y aprendizaje.
2. Desarrollo de un prototipo basado en el Arduino.
3. Estudio de la WRAPSTER (fusión entre una flejadora y una envolvedora de palets).
4. Implicación en el desarrollo de la WRAPSTER.

Finalmente vamos a definir los **objetivos** marcados para este proyecto:

- Familiarización y aprendizaje con la plataforma Arduino.
- Integración de la red de sensores y actuadores de la plataforma.
- Evaluación de la Shield RFID como sistema de identificación.
- Evaluación de la Shield GPRS/GSM como sistema de comunicación.
- Evaluación de un sistema de monitorización telemática.
- Desarrollo de un prototipo.
- Colaboración entre empresas.
- Desarrollo de la WRAPSTER.
- Encontrar soluciones y gestionar siguientes versiones.

El objetivo de este proyecto se puede resumir por tanto en hacer el diseño de una plataforma compuesta por la placa Arduino, una shield GPRS/GSM, una shield RFID y una serie de actuadores que sirvan de ejemplo de un funcionamiento real como, por ejemplo: el inicio de un ciclo de enfiado o un sistema de alerta y alarma del sistema. El cual enviaría una alerta vía SMS indicando la alarma y el estado de la máquina. A su vez, se pretende implementar otra aplicación que sea capaz de trabajar de manera desatendida ('standalone'), y que responda ante mensajes de texto para consultar el estado de la máquina en cualquier momento o regular el consumo acorde a las instrucciones que el usuario le transmita vía SMS.

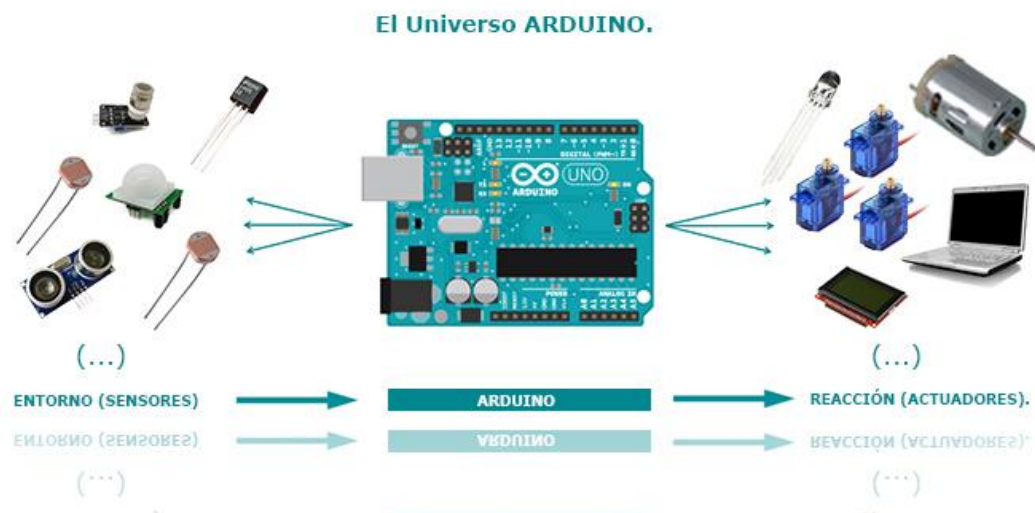
Todos estos objetivos serán siempre con vistas al desarrollo de una mejora del sistema que se llevará a cabo en la empresa.

1.4 JUSTIFICACIONES

- **Arduino como base del proyecto:**

Arduino es una compañía de hardware libre y una comunidad tecnológica que diseña y manufactura placas de desarrollo de hardware y software. Estando compuestas, respectivamente, por circuitos impresos que integran un microcontrolador y un entorno de desarrollo (IDE) para programar cada placa. Por tanto, se trata de una herramienta para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Por otra parte, como se puede ver en la ilustración 1, es el corazón de cualquier sistema formado por sensores y actuadores, es el microcontrolador, de tal forma que, una vez definido un conjunto de estos dispositivos requeridos en cualquier aplicación, se puede programar en la placa un código que permita automatizar cualquier proceso. Así, como también aporta gran versatilidad y flexibilidad, se pueden aportar soluciones tecnológicas en cualquier ámbito.



- **Sistema de comunicación GRPS/GSM:**

La expansión de las redes de telefonía móvil a lo largo del territorio nacional ha abierto la puerta a una alternativa eficiente y de bajo costo para el monitoreo y control de equipamiento ubicado en lugares remotos. Para monitorear variables en lugares remotos y de difícil acceso, podemos utilizar como canal de comunicación la red de telefonía celular, accediendo a ella por medio de la tecnología GPRS o HSDPA y utilizando los equipos adecuados para realizar dicha conexión. Imaginemos un proyecto (como es el presente caso) en el que se quiera monitorizar el estado y/o funcionamiento de una maquina dentro de una fábrica, un lugar donde se puede modificar el entorno de trabajo dependiendo de la producción, donde un cable puede presentar un problema y donde el Wi-Fi no acaba de llegar.

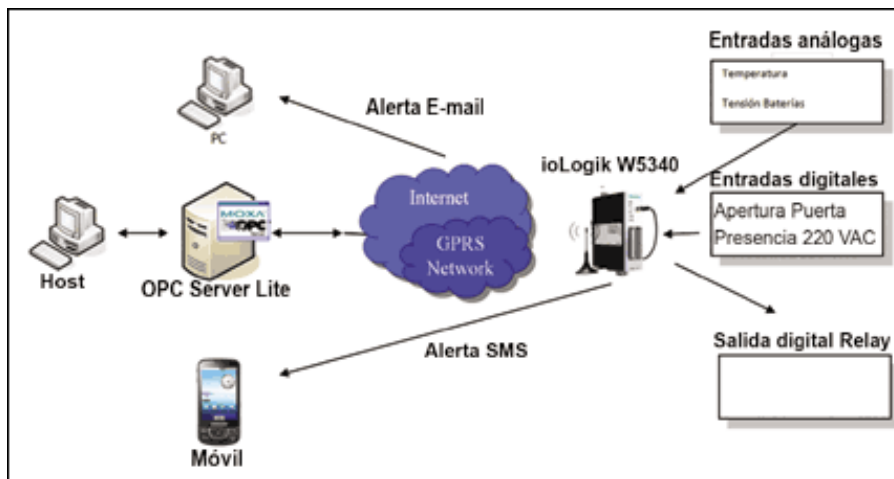


Ilustración 2 Sistema de telecomunicaciones

Como podemos apreciar en la Ilustración 2 los servicios GSM/GPRS nos permiten realizar sistemas inteligentes los cuales podemos usar para monitorear remotamente desde cualquier dispositivo conectado a la red, además de realizar funciones automáticas, se pueden tener todos los sensores y actuadores que se necesiten, también se instalan en casi cualquier parte, sin necesidad de cables. Podemos encontrar toda la información sobre estos sistemas en internet, como funcionan y trabajan. (universia, 2005)

- **Sistema de identificación por RFID:**

La tecnología RFID (Radio Frequency Identification) es un sistema de almacenamiento de datos remotos que usa dispositivos conocidos mayormente como etiquetas o tags RFID.

El propósito fundamental de esta tecnología es ser capaz de obtener el identificador de un elemento mediante ondas de radio.

Las etiquetas RFID pueden ser unas etiquetas normales (normalmente etiquetas autoadhesivas) a las que se les introduce un “inlay” RFID. Estos “inlays” contienen antenas para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Las etiquetas suelen ser pasivas, que significa que no necesitan alimentación eléctrica interna.

El uso de esta tecnología no es nuevo en la industria, sino que hace años que se utiliza en logística a modo de localización y albarán de materiales como se puede apreciar en la Ilustración 3. Dentro de la Web “futuros” podemos encontrar diferentes aplicaciones de esta tecnología. (FUTUROS, 2012)

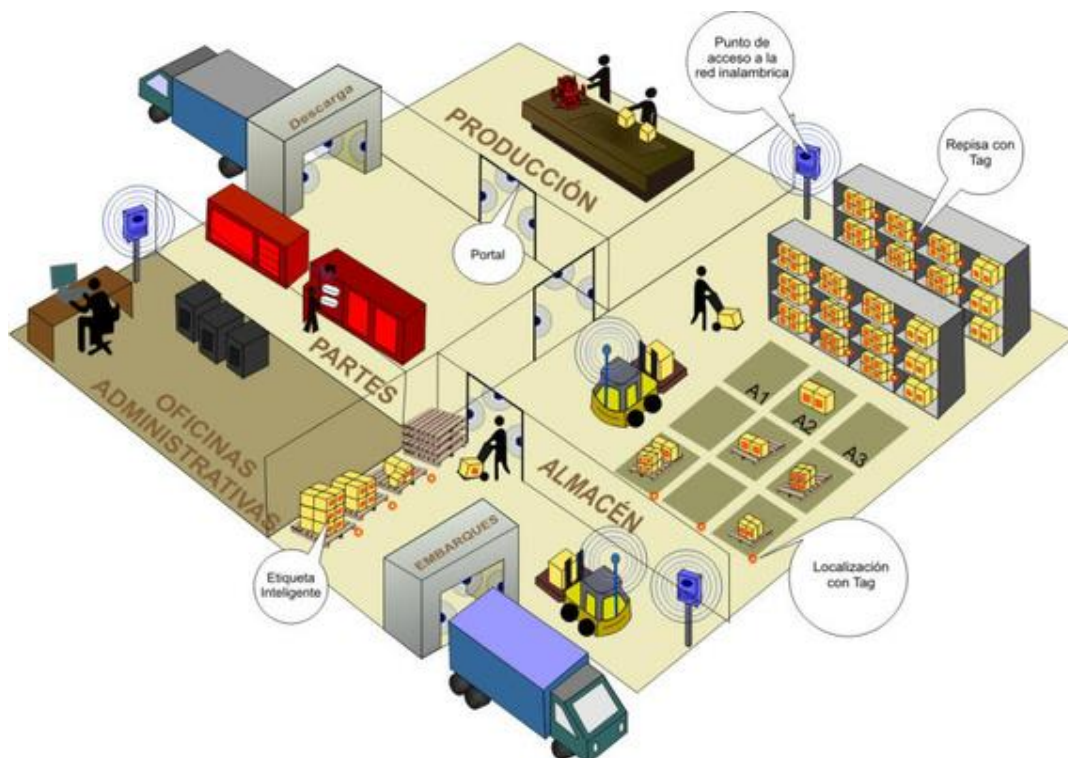


Ilustración 3 Sistema RFID en empresa

Al ser una tecnología que no requiere de cables ni un contacto visual directo, hace que el RFID sea la mejor solución al identificador de bobinas.

1.5 ANTECEDENTES, EL R-CONNECT

La multinacional ROBOPAC es una empresa de Santorini que se dedica a la fabricación de maquinaria industrial dedicada al sector de Packaging.

“Innovación y conexión para el servicio al cliente: la tranquilidad de la máxima productividad como resultado del monitoreo constante de la operación adecuada de la máquina.

Robopac lanza R-Connect, un sistema de monitoreo tecnológicamente avanzado que combina innovación y conexión, lo que permite un control total del funcionamiento de la máquina y un soporte total para el cliente final. Esta es una innovación pionera en el campo de las máquinas semiautomáticas, que Robopac pone a disposición de los distribuidores y especialmente de los usuarios finales, pero también proporciona una actualización del sistema que se puede aplicar en máquinas más avanzadas.” (ROBOPAC R-Connect, 2018)

Control Pack a raíz de este innovador sistema lanzado por ROBOPAC, quiere realizar su propio sistema basado en el R-connect. Poder realizar una “caja negra” que se pueda aplicar a cualquier máquina automática o semi automática para la captación de datos y poderlos transmitir, junto con una página web y/o aplicación, se podrá hacer el seguimiento, control y gestión de las máquinas con el sistema SPK-Connect.

El R-Connect únicamente hace un seguimiento de los parámetros de la máquina, es decir puede visualizar como está configurada la máquina, si está operativa, su historial de errores y donde está ubicada la máquina.

El SPK-Connect pretende ir un paso más lejos añadiendo el control de funcionamiento por tag y el cálculo de consumible.



Ilustración 4 R-Connect

1.6 SUMARIO CONTENIDO DEL PROYECTO

El proyecto se divide en cuatro grandes apartados que abarcarán las diferentes fases por las que se ha tenido que pasar para el desarrollo de este.

En primer lugar, llevamos a cabo un pequeño estudio teórico sobre el estado del arte de las diferentes tecnologías que guardan alguna relación con el proyecto, como son el concepto de Internet de las Cosas, los sistemas de comunicaciones móviles GSM y GPRS, y los principales comandos AT. Todo ello quedará resumido más adelante.

Tras ello, se evaluará todo lo relacionado con la base del proyecto, es decir, con la plataforma Arduino (placas, shields de extensión, sensores, actuadores, etc.). Comenzaremos analizando los diferentes modelos de placas Arduino disponibles en el mercado, con el fin de escoger el que mejor se adapte a las necesidades del sistema que se tiene como objetivo desarrollar. A continuación, examinaremos los módulos de expansión (shields) compatibles con esta plataforma, que nos permitan extender sus capacidades y contar con un dispositivo capaz de establecer un medio de comunicación a través de la tecnología GPRS/GSM y la identificación por tag.

En el siguiente apartado, se expondrán con detalle todas las tareas realizadas en el desarrollo experimental del proyecto, como las pruebas con Arduino, pruebas de envío y recepción de SMS a través del módulo GPRS/GSM, etc.

Por último, veremos implementación dentro de la industria del SPK-Connect en una máquina real, la WRAPSTER.

Finalmente, dejaremos un apartado para expresar las conclusiones que se han podido sacar tras el resultado final del proyecto.

1.7 PLANIFICACIÓN

A la hora de realizar la planificación se ha tenido en cuenta las dificultades para congeniar el trabajo diario dentro de la empresa y las clases en la universidad con el propio trabajo de fin de grado, como es un proyecto de la empresa, es totalmente imposible llevar a cabo los plazos a tiempo a causa de todas las variables y problemas que pueden surgir. Hay que tener en cuenta que en este proyecto empresarial hay diferente gente involucrada y las tomas de decisiones son lentas.

A fin de poder presentar este trabajo a la universidad, he tenido que realizarlo acorde con los intereses de la empresa y los míos propios, con lo cual tuve que solicitar un aplazamiento del TFG a fin de poder disponer de la información requerida para el último capítulo de esta memoria, la WRAPSTER.

Considero muy importante este último capítulo puesto que consolida todos los capítulos anteriores y deja al descubierto las intenciones y viabilidad del proyecto al llevar a cabo una versión temprana del SPK-Connect.

Poder trabajar en la WRAPSTER, una máquina que combina la flejada con la envoltura de palets, con un sistema de gestión de datos telemáticos es de gran relevancia.

CAPITULO II. ESTADO DEL ARTE

2.1 INDUSTRIA 4.0

El concepto de Industria 4.0 se refiere a la inminente cuarta revolución industrial o la ciber industria del futuro que corresponde a una manera nueva de organizar los medios de producción. Como bien se expone en el portal IDIa, donde hace un seguimiento de las revoluciones industriales, como podemos ver en la ilustración 5, y los avances tecnológicos. (IDIa, 2016).

El objetivo que se pretende alcanzar es la puesta en marcha de un gran número de fábricas inteligentes capaces de una mayor adaptación a las necesidades y a los procesos de producción, así como una mayor eficiencia de los recursos utilizados. La Industria 4.0 es el Lean Managemet en estado puro.

Hoy en día el planteamiento que nos hace la Industria 4.0, nos deja unos conceptos muy importantes:

- 1- Una capacidad de adaptación constante a la demanda.
- 2- Servir al cliente de una forma más personalizada (valor añadido).
- 3- Aporta un servicio post venta más directo con el cliente.
- 4- Diseñar, producir y vender productos en menos tiempo.
- 5- Añadir complementos de supervisión a elementos físicos.
- 6- Crear series de producción más cortas y rentables.
- 7- Aprovechar la información para su análisis desde múltiples canales donde ser capaces de analizarla en tiempo real.

Para poder llevar a cabo una integración total de la Industria 4.0, se tendrá que incidir en unos puntos claves.

- 1- Big data y análisis de datos.
- 2- Cloud Computing.
- 3- Ciberseguridad.
- 4- Robótica.
- 5- Internet de las cosas.
- 6- Simulación y prototipo.
- 7- Realidad aumentada.
- 8- Cultura (adaptación a los cambios)
- 9- Integración de procesos.

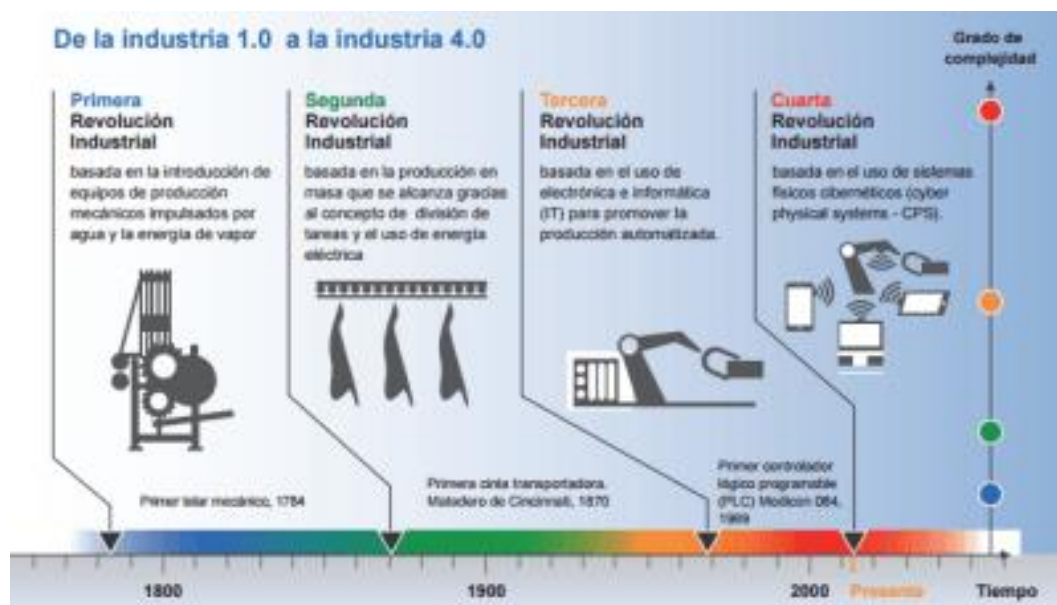


Ilustración 5 Revoluciones Industriales

2.1.1 EL INTERNET DE LAS COSAS

El Internet de las cosas es un concepto que se refiere a la interconexión de objetos del día a día (electrodomésticos, ropa, libros, productos alimenticios, etc.) con internet. El concepto del Internet de las cosas fue propuesto por Kevin Ashton en el Auto-ID Center del MIT (Instituto tecnológico de Massachusetts) en 1999, donde se realizaban investigaciones en el campo de la identificación por radiofrecuencia en red (RFID). (El PAIS, 2007).

En la ilustración 6 se resume el concepto del Internet de las cosas...

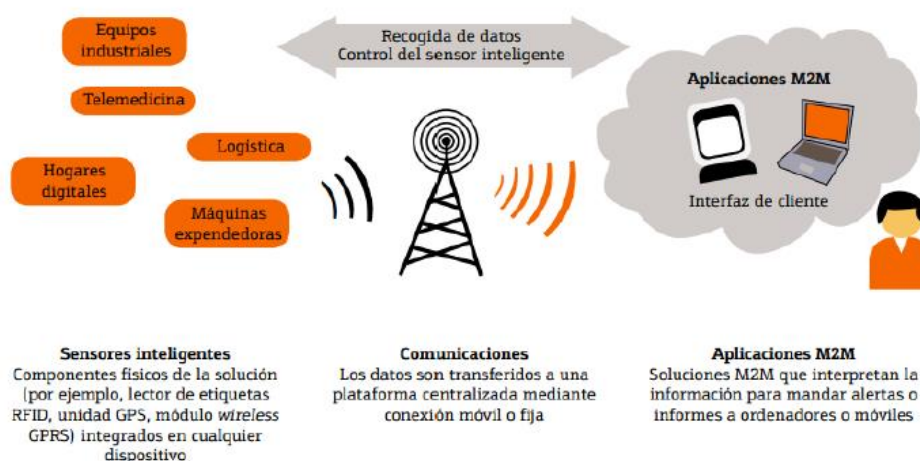


Ilustración 6 Concepto del "Internet de las cosas"

Las utilidades del Internet de las Cosas pueden considerarse prácticamente infinitas. Cada vez son más los medios que tenemos a nuestra disposición para implementar, de manera no muy compleja, un posible dispositivo capaz de dotar a ciertos objetos de una conexión a la red ya sea por medio de una tecnología u otra.

La llegada de IPv6 supone un factor clave en el desarrollo del concepto IoT. Gracias a este nuevo protocolo (diseñado para reemplazar a IPv4) se evitará que el crecimiento de Internet quede restringido, y hará posible la gestión de direccionamiento de innumerables dispositivos.

Por otro lado, ya son muchas las aplicaciones móviles y servicios en la nube que nos permiten la conexión a todos estos dispositivos, y proporcionan una vía para el tratamiento de una inmensa cantidad de datos en tiempo real (sistemas 'Big Data'),

facilitando la integración de infinidad de sensores aplicables prácticamente a cualquier tipo de necesidad. De hecho, ya han surgido incluso redes sociales de sensores, como la plataforma ‘Xively’ (Xively, 2018), donde los usuarios comparten datos en tiempo real procedentes de distintos sensores. En los próximos años se espera un gran aumento en el número de equipos de uso cotidiano interconectados como comenta el portal Kurzweil 2025 (Kurzweil, 2011), entre otras cosas, gracias a la inminente llegada de todos estos sensores inteligentes a nuestros hogares. En la ilustración 7 se muestra un gráfico con las expectativas de crecimiento en vistas al año.

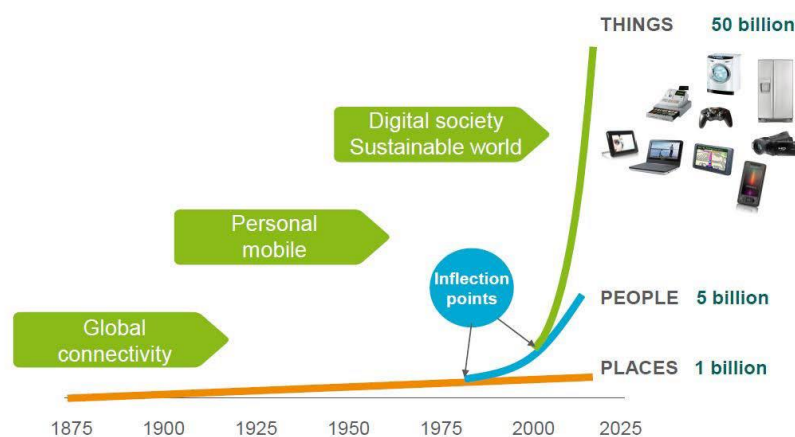


Ilustración 7 Expectativas de crecimiento de dispositivos inteligentes interconectados

También son muchas las empresas que ofrecen, o están interesadas, en soluciones IoT. La gestión de recursos y la eficiencia energética son las aplicaciones más solicitadas. Tecnologías inalámbricas como las redes móviles, WIFI, Zigbee, Bluetooth, etc., permiten optimizar posibles soluciones y facilitan su despliegue.

Sin ir más lejos, Arduino ha marcado un punto de inflexión en este sentido, convirtiéndose en la herramienta ideal para llevar a cabo multitud de prototipos.

Gracias a su bajo coste, sencillez y a la variedad de modelos que podemos encontrar, resulta una herramienta de gran ayuda a la hora de implementar ideas y soluciones de ámbito doméstico. Además, existen multitud de sensores y actuadores compatibles con esta plataforma, mediante los cuales podemos recopilar datos de nuestro entorno, analizarlos, y actuar en consecuencia, incluso conectar con otros dispositivos a través de las distintas tecnologías de comunicación (GSM/GPRS, 3G, Bluetooth, RFID, etc.) aprovechando toda una variedad de shields de expansión.

1.3 GSM

En internet podemos encontrar información muy destacable sobre este sistema de comunicación tan cotidiano, en *digitaltrends* (digitaltrends, 2017) nos cuentan que, GSM es la abreviatura de ‘Sistema Global para las comunicaciones Móviles’ (en inglés, Global System for Mobile communications). A comienzos del siglo XXI, es el estándar más utilizado de Europa. Conocido como estándar de segunda generación (2G), su principal diferencia respecto a la primera generación de teléfonos móviles es que sus comunicaciones son totalmente digitales.

El estándar GSM fue desarrollado a partir de 1982, cuando fue estandarizado por primera vez, denominado "Groupe Spécial Mobile". Surgió como idea para el desarrollo de un estándar europeo de telefonía móvil digital. En 1991 se convirtió en un estándar internacional llamado "Sistema Global de Comunicaciones Móviles", y comenzaron a presentarse los primeros prototipos de telefonía GSM.

En Europa, el sistema GSM utiliza las bandas de frecuencia de 850, 900 y 1800 MHz, mientras que en los Estados Unidos se usa la banda de frecuencia de 1900 MHz. En consecuencia, los dispositivos de comunicaciones móviles que pueden operar tanto en Europa como en Estados Unidos se conocen como cuatribanda (Quadband). El estándar GSM permite transmisiones digitales de voz y datos, como mensajes de texto (SMS) o mensajes multimedia (MMS).

Respecto a su arquitectura de red, en GSM todo terminal móvil debe estar constituido por una tarjeta SIM (Módulo de identificación de abonado) y el propio dispositivo, normalmente un teléfono móvil. La tarjeta SIM es la encargada de identificar en la red al usuario y al terminal móvil. Estos dispositivos se identifican gracias a un número exclusivo de identificación denominado IMEI (Identificador internacional de equipos móviles), compuesto por 15 dígitos. Por otro lado, cada tarjeta SIM también posee un número de identificación único denominado IMSI (Identificador internacional de abonados móviles).

En la ilustración 8 podemos ver la arquitectura de red correspondiente al sistema GSM. Está compuesta por múltiples estaciones base (BTS), que, a su vez, se conectan a un controlador de estaciones base (BSC), encargado de la administración de la red. A este sistema compuesto por el BSC y sus correspondientes estaciones base conectadas al mismo, se le conoce como BSS (Subsistema de estaciones base).

En un nivel superior estarían los Centros de conmutación móvil (MSC), al que se conectan físicamente los controladores de estaciones base. Éste es el encargado de establecer la conexión con la red de telefonía pública y con Internet. Como bien nos explica el portal movilfacil.com (movilfacil, 2011).

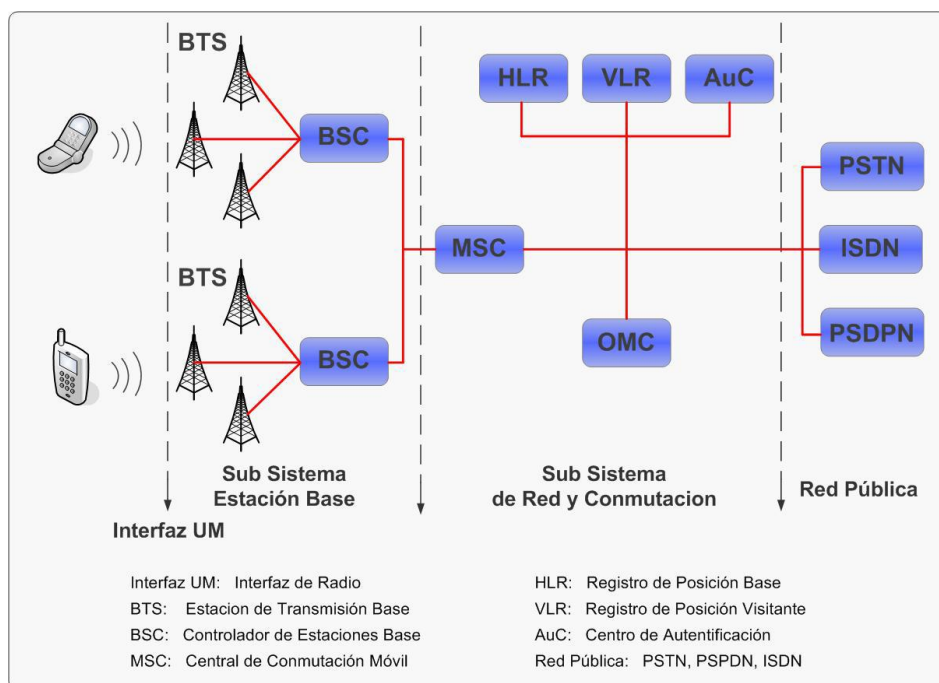


Ilustración 8 Arquitectura de red del sistema GSM

2.3 GPRS

El estándar GPRS o Servicio General de Paquetes vía Radio (en inglés, General Packet Radio Service) es una evolución del sistema GSM. Es también conocido como GSM++, pero dado que se trata de un estándar de telefonía móvil intermedio entre la segunda generación (2G) y la tercera (3G), a menudo recibe la nomenclatura de 2.5G como bien nos explican en el portal web movilfacil (movilfacil, 2011). GPRS extiende la arquitectura del estándar GSM para permitir la transferencia de datos mediante conmutación de paquetes con velocidades de transferencia que rondan los 114 Kbps.

Al contrario de lo que ocurre en conmutación de circuitos, en el estándar GPRS, gracias a su modo de transferencia de paquetes, las transmisiones de datos sólo utilizan la red cuando es necesario, permitiendo la tarificación por volumen de información transmitida en lugar de por tiempo de conexión, por lo tanto, el usuario puede permanecer conectado sin costo adicional, ya que sólo utilizará la red cuando envíe o reciba un paquete de datos.

Para el acceso a la red de datos, el estándar GPRS utiliza el protocolo IP, mientras que, para el transporte de voz, emplea la arquitectura de la red GSM.

A parte de actualizar algunos servicios con los que ya contaba GSM, la tecnología GPRS admite otra serie de características que no estaban disponibles en 2G:

- Servicios de mensajes cortos (SMS)
- Servicios de mensajes multimedia (MMS)
- Servicio punto a punto (PTP); para la conexión cliente-servidor en una red IP
- Servicio punto a multipunto (PTMP); para el envío de multidifusión.

En la ilustración 9 se muestra la estructura funcional del sistema GPRS, basada en la adición de nuevos nodos sobre la infraestructura correspondiente a GSM. A dichos nodos se les conoce como GSN (nodos de soporte GPRS):

- El SGSN (Nodo de soporte de servicios GPRS) se encarga del encaminamiento de los paquetes IP entrantes y salientes dirigidos hacia, o procedentes, de cualquier abonado GPRS situado dentro de la zona geográfica a la que da servicio ese SGSN.
- EL GGSN (Nodo de soporte pasarela GPRS) sirve de interfaz hacia las redes externas de paquetes IP. Encamina las direcciones IP de los abonados servidos por la red GPRS, intercambiando información de encaminamiento con la red externa. El GGSN prepara la comunicación con redes externas y gestiona las sesiones de GPRS. También incluye funciones para asociar abonados con la SGSN que corresponda. También recopila datos de tarificación y uso de la red de datos externa.

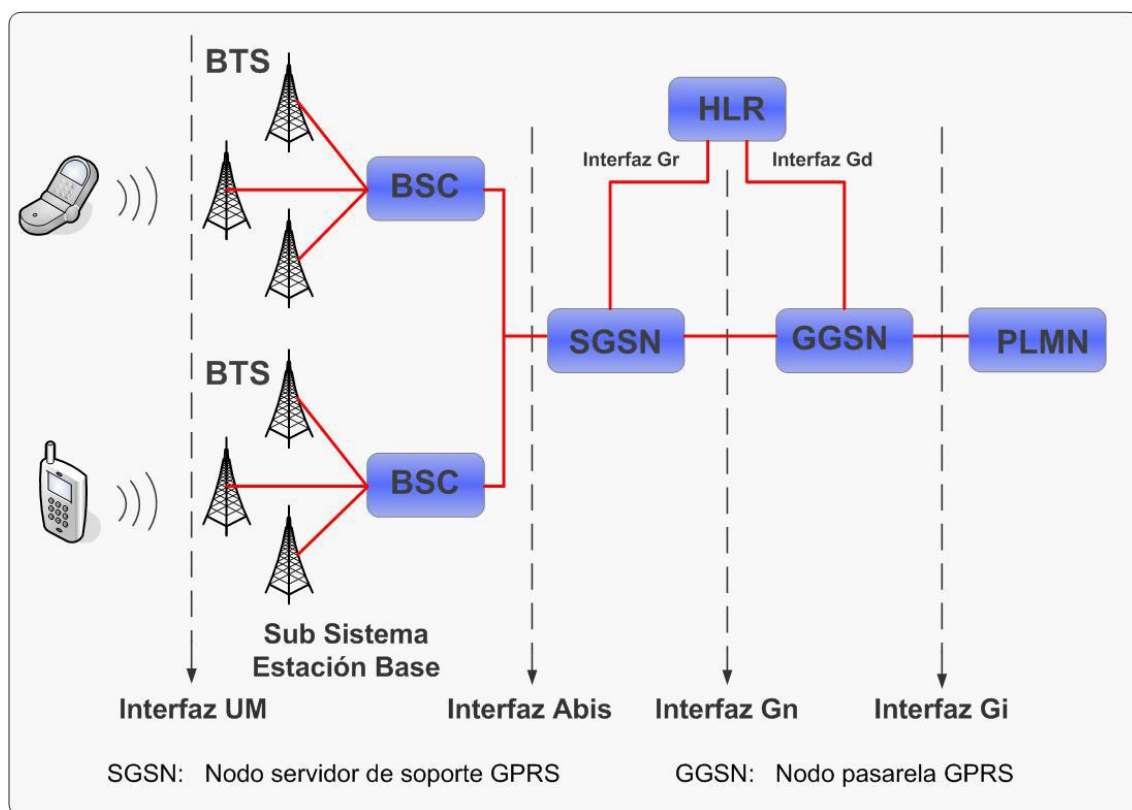


Ilustración 9 Arquitectura de red del sistema GPRS

2.4 COMANDOS AT

Los comandos AT, también conocidos como comandos Hayes (en honor a su desarrollador Dennis Hayes), son una serie de instrucciones que conforman un interfaz de comunicación entre usuario y modem. Su abreviatura AT por la que son mundialmente conocidos estos comandos proviene de la palabra ‘attention’. Como bien nos explican en la Web para desarrolladores de *Panotec* donde podemos encontrar ejemplos de los comandos AT (Prometec, 2018).

Aunque la finalidad principal de los comandos AT fue la comunicación con módems, la telefonía móvil GSM/GPRS también adoptó este lenguaje como estándar de comunicación.

En la actualidad, todos los terminales móviles GSM poseen una serie específica de comandos AT que nos permiten configurarlos por medio de estas instrucciones e indicarles una serie de acciones que queremos que ejecuten, tales como marcar un número de teléfono, enviar o leer un SMS, consultar el estado de conexión a la red, leer o escribir en la agenda de contactos, etc.

Gracias a que la transmisión de comandos AT no depende del canal de comunicación a través del cual estos sean enviados (cable, infrarrojos, Bluetooth, etc.), podremos utilizar nuestra placa Arduino para transmitir dichos comandos a un módulo GPRS/GSM que sea capaz de interpretarlos y actuar en consecuencia.

Como son muchos los comandos existentes, véase el manual de comandos AT oficial de nuestro módulo GSM/GPRS (SIM900) (Manual comandos AT, 2010), únicamente vamos a mencionar junto con una breve descripción, aquellos que puedan resultarnos de principal interés en el desarrollo del proyecto:

- **AT** – Con este comando verificaremos que el módulo está recibiendo nuestras instrucciones. Si todo es correcto, tras enviarlo debe responder con un “OK”. En caso contrario no obtendremos respuesta alguna.
- **AT+CPIN?** – Utilizaremos esta instrucción para comprobar si nuestra tarjeta SIM está desbloqueada o si, por el contrario, requiere introducir el código PIN para proceder con el desbloqueo de la misma. Cuando la SIM esté operativa responderá con un “ready”.

- **AT+CPIN="****"** – En el caso de que necesitemos introducir el PIN, éste es el comando que debemos enviar, escribiendo los 4 dígitos correspondientes al código de desbloqueo en el lugar de los asteriscos, delimitado entre comillas.
- **AT+CREG?** – Con este comando estamos preguntando por el estado de conexión a la red. La respuesta recibida seguirá la siguiente notación: +CREG: <n>,<stat>, donde:

<stat> = 0 → No registrado, no está buscando una conexión de red

<stat> = 1 → Registrado a la red nacional

<stat> = 2 → No registrado, pero buscando la red

<stat> = 3 → Registro denegado

<stat> = 5 → Registro tipo roaming

- **AT+COPS?** – Mediante esta instrucción recibiremos la confirmación de la compañía en la que está registrada nuestra tarjeta SIM (Movistar, Orange, Vodafone, etc.)

- **AT+CMGF=<f>** – Selecciona el formato del mensaje SMS, donde:

<f> = 0 → modo PDU

<f> = 1 → modo texto

- **AT+CSCS="IRA"** – Con este comando seleccionamos el set de caracteres que es utilizado para el envío de SMS. En nuestro caso nos interesa configurar el módulo en modo "IRA" (International Reference Alphabet).
- **AT+CMGS="num_movil"** – A través de este comando marcaremos el número de móvil al que queremos hacer llegar el SMS. Irá delimitado por comillas.
- **AT+CMGR=<r>** – Es el comando utilizado para leer SMS almacenados en la memoria de la tarjeta SIM. En lugar de <r> pondremos el número correspondiente a la posición del mensaje que queremos leer. Por ejemplo, si queremos leer el último mensaje recibido, pondremos un '1', si queremos leer el penúltimo mensaje pondremos un '2', y así sucesivamente.
- **AT+CPMS="SM"** – Con esta instrucción seleccionamos el directorio del que queremos leer un mensaje de texto. En función de las siglas que pongamos tras el igual, podemos recopilar la información de distintas memorias. En este caso, con las siglas "SM", seleccionamos como directorio la memoria de la tarjeta SIM.

- **AT+CMGD=<n>** – Este comando sirve para eliminar SMS de la memoria de la tarjeta SIM. Sustituiremos <n> por la posición en la memoria que ocupe el SMS que queremos borrar. Por ejemplo, para borrar el primer mensaje almacenado en la memoria de la SIM la instrucción sería AT+CMGD=1.
- **AT+CMGL="ALL"** – Con este comando podemos ver en una lista todos los SMS que tengamos en la SIM, tanto leídos, como no leídos.
- **+CMTI: "SM", <pos>** – Esta es la instrucción que recibiremos automáticamente por parte del módulo cuando se reciba un SMS, donde <pos> es el número correspondiente a la posición en memoria en la que se ha almacenado dicho mensaje. Por ejemplo, si tenemos la memoria de la SIM vacía y nos llega un SMS, la instrucción que nos enviaría el módulo sería +CMTI: "SM", 1.
- **AT+CGATT=1** – Esta es la función para conectarnos a la red GPRS.
- **AT+CSTT="Nombre del Punto de Acceso o APN","usuario","contraseña"**
– Esta es la manera de tener acceso a internet, el APN usuario y contraseña va en función de la compañía de la tarjeta SIM, estos datos se encuentran fácilmente en internet haciendo una pequeña búsqueda.
- **AT+CIICR** – Activamos el perfil de datos inalámbricos.
- **AT+CIFSR** – Obtenemos nuestra dirección IP y la guardamos
- **AT+CIPSTART**– Comando para indicar la dirección IP y el puerto a la que queremos hacer la conexión.
- **AT+CIPSEND** – Comando para enviar datos al enlace señalado.

2.5 RFID

La Identificación por Radiofrecuencia, a partir de ahora RFID, es un término genérico que define una serie de tecnologías que emplean ondas de radio para la identificación automática de personas u objetos. Una de las claves es que la recuperación de la información contenida en la etiqueta sin necesidad de cables ni contacto físico ni visual. Como bien nos explican en el portal Prometec, donde podemos encontrar diferentes proyectos Arduino que utilizan esta tecnología (Prometec, 2016).

Se trata de una tecnología de captura e identificación automática de información contenida en etiquetas electrónicas. Cuando estas etiquetas entran dentro de la cobertura de un lector, éste envía una señal para que dicha etiqueta le transmita la información que tenga almacenada en su memoria interna, tal y como se muestra en el esquema general de funcionamiento del sistema RFID de la ilustración 10.

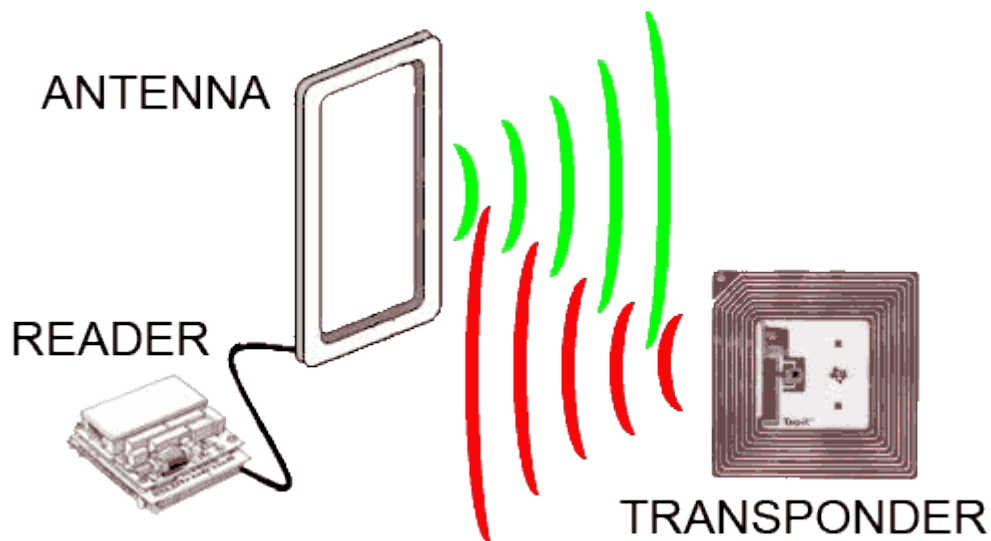


Ilustración 10 Sistema de Radio Frecuencia

Las etiquetas RFID están disponibles en una gran variedad de formatos, tales como pegatinas adheribles, tarjetas, llaveros, pueden integrarse en un determinado producto o, incluso, insertarse bajo la piel en un animal o humano.

Los RFID son ampliamente empleados, por ejemplo, en sistemas de alarma, aplicaciones comerciales en sustitución de códigos de barras, cerraduras electrónicas, sistemas de pago, tarjetas personales, control de accesos recintos como gimnasios o piscinas, fichaje en empresas, entre otras muchas aplicaciones.

El RFID es un conjunto de tecnologías inalámbricas diseñadas para obtener una información almacenada en un dispositivo denominado etiqueta (tag).

El lector (transceptor) es en realidad un emisor-receptor que, en primer lugar, emite una señal para iniciar la comunicación con las etiquetas (transpondedores). Esta señal es captada por las etiquetas dentro del alcance, la cual responden transmitiendo la información que, almacenada, finalmente, es captada y decodificada por el lector RFID.

El RFID puede operar en cuatro bandas de frecuencia, siendo la más frecuente 13.56 Mhz.

- Baja frecuencia 125-134.2 kHz. Control de animales, llaves de automóviles...
- Alta frecuencia 13.56 MHz. Control de accesos, control de artículos en tiendas...
- Ultra alta frecuencia (UHF) 868 – 956 GHz
- Microondas, 2,45 GHz

Existen etiquetas RFID de sólo lectura, es decir, en las que la información que contienen es grabada durante su fabricación y no puede modificarse, y etiquetas de lectura y escritura, en las que podemos sobrescribir la información de la etiqueta.

Respecto a la alimentación, existen etiquetas RFID activas que disponen de su propia fuente de energía (por ejemplo, una batería). El rango de lectura puede ser de 10m a 100m.

Por contra las etiquetas RFID pasivas obtienen su energía por inducción de la onda electromagnética emitida por el lector. Por tanto, no requieren fuente de alimentación. Sin embargo, el alcance de lectura se reduce a unos centímetros.

2.6 PLATAFORMA ARDUINO

A continuación, entramos en detalle sobre los distintos elementos que conformarán el sistema basado en la plataforma Arduino. Comenzaremos con una breve presentación de lo que es en sí esta plataforma, también se evaluarán las distintas shields (módulos de expansión).

¿Qué es Arduino? - Arduino es una plataforma electrónica de hardware libre basada en una placa con un microcontrolador. Con software y hardware flexibles y fáciles de utilizar, Arduino ha sido diseñado para adaptarse a las necesidades de todo tipo de público, desde aficionados, hasta expertos en robótica o equipos electrónicos. También consta de un simple, pero completo, entorno de desarrollo, que nos permite interactuar con la plataforma de manera muy sencilla. Se puede definir por tanto como una sencilla herramienta de contribución a la creación de prototipos, entornos, u objetos interactivos destinados a proyectos multidisciplinarios y multitecnológicos.

En la ilustración 11 podemos ver una de sus placas más vendidas a nivel mundial, se trata del modelo Arduino UNO.

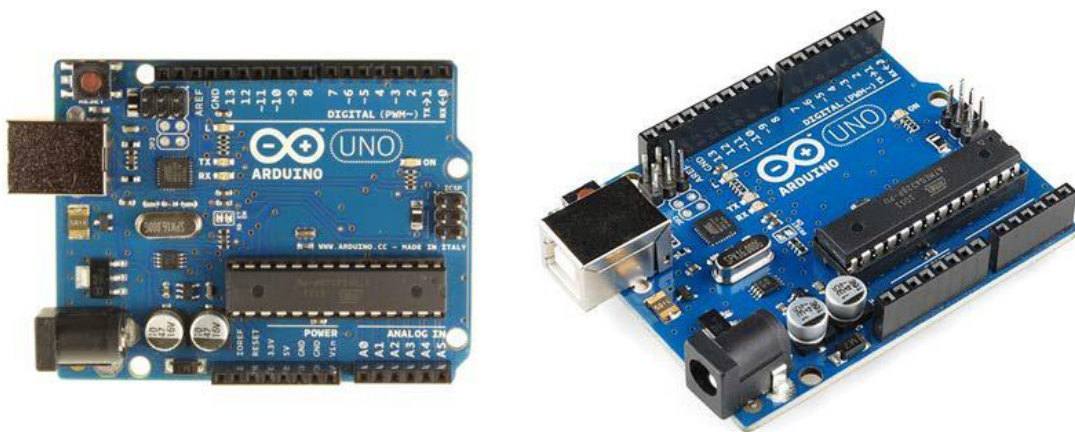


Ilustración 11 Arduino UNO

A través de Arduino podemos recopilar multitud de información del entorno sin excesiva complejidad. Gracias a sus pines de entrada, nos permite jugar con toda una gama de sensores (temperatura, luminosidad, presión, etc.) que nos brindan la capacidad de

controlar o actuar sobre ciertos factores del entorno que le rodean, como, por ejemplo: controlando luces, accionando motores, activando alarmas...y muchos otros actuadores.

Gracias a que la plataforma es de hardware libre, las placas Arduino pueden ser hechas a mano por cualquier aficionado o compradas ya montadas de fábrica. Respecto al software, es totalmente gratuito, y está disponible para la descarga por cualquier interesado en la propia página web de (Arduino, 2018). El entorno de desarrollo dispone de un propio lenguaje de programación para el microcontrolador de la placa Arduino, basado en Processing/Wiring.

Uno de los principales motivos por el cual resulta muy interesante la utilización de la plataforma Arduino para determinados proyectos, se basa en su independencia respecto a tener que mantenerse conectado a un PC. Arduino es perfectamente capaz de trabajar en modo ‘standalone’, solo es necesario asegurarnos de haber cargado previamente el programa que deseamos que mantenga en ejecución. Si bien, todo esto no le priva de poder operar manteniendo en todo momento la conexión con el PC, siendo capaz de comunicarse con diferentes tipos de software, como, por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data, etc.

Un factor de éxito de Arduino ha sido la comunidad que está apoyando este proyecto y que día a día publica nuevos contenidos, divulga y responde dudas.

En internet hay disponible todo tipo de cursos, tutoriales, herramientas, proyectos, etc. que ayudan a que se pueda usar Arduino con facilidad.

La placa escogida para realizar este proyecto es la versión UNO R3, la cual puede verse a continuación. Se ha elegido esta placa simplemente porque es la que se recomienda para principiantes en el mundo de Arduino, además dispone de las entradas y salidas digitales perfectas para este proyecto.

2.6.1 ENTORNO DE DESARROLLO

El entorno de desarrollo en Arduino (IDE) es el encargado de la gestión de la conexión entre el PC y el hardware de Arduino con el fin de establecer una comunicación entre ellos por medio de la carga de programas. Como podemos ver en la ilustración 12 el IDE de Arduino se compone de:

- Un **editor de texto**. - donde escribir el código del programa.
- Un **área de mensajes**. - a través del cual el usuario tendrá constancia en todo momento de los procesos que se encuentren en ejecución, errores en código, problemas de comunicación, etc.
- Una **consola de texto**. - mediante la que podremos comunicarnos con el hardware Arduino y viceversa.
- Una **barra de herramientas**. - donde podremos acceder a una serie de menús y a los botones con acceso directo a las principales funcionalidades de Arduino



Ilustración 12 IDE del Arduino

A través de la IDE de Arduino, podemos escribir el código del programa y crear lo que se conoce por "sketch" (programa). ¿Por qué lo llamamos sketch y no programa? Pues porque el IDE de Arduino viene de Processing, y en este lenguaje de programación enfocado al mundo gráfico, cada código es considerado un boceto, en inglés "sketch".

El sketch permite la comunicación con la placa Arduino. Estos programas son escritos en el editor de texto, el cual admite las posibilidades de cortar, pegar, buscar y remplazar texto.

En el área de mensajes se muestra, tanto la información mientras se cargan los programas, como los posibles errores que tengamos a la hora de compilar, ya sea por problemas en el código del sketch, por fallo en la detección de nuestro Arduino en el puerto USB, o por cualquier otro problema que sea detectado.

La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones.

Desde la barra de herramientas tenemos acceso directo a las principales funcionalidades que ofrece el IDE de Arduino, como, por ejemplo: verificar el proceso de carga, crear un nuevo sketch, abrir un sketch ya existente, guardar los programas, abrir el Monitor Serial, etc.

Monitor serie

El monitor serie es una de las partes más importantes del IDE de Arduino porque es nuestra ventana para la comunicación entre Arduino y el ordenador, que se hace a través del cable USB.

El monitor serie muestra los datos enviados por el Arduino a través del puerto serie también nos permite mandar datos al Arduino mediante el puerto serie. En la ilustración 13 podemos ver la ventana de comunicación.

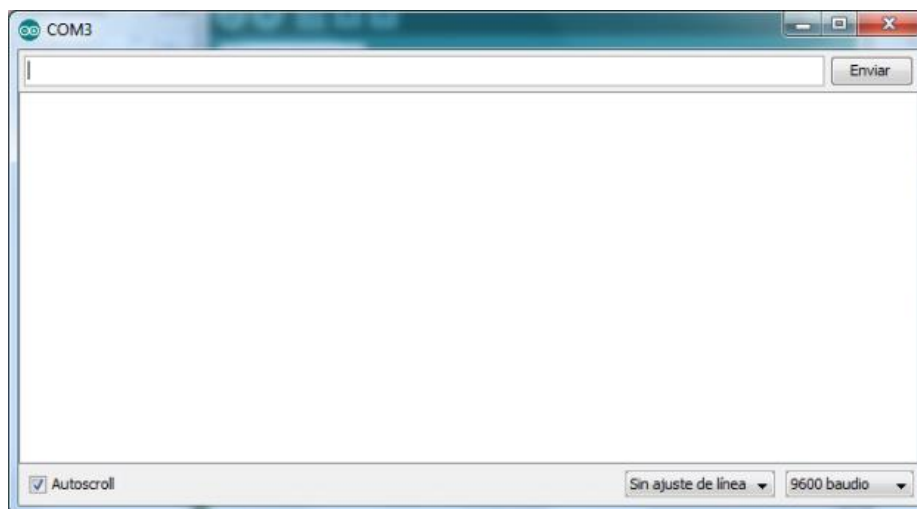


Ilustración 13 Monitor Serie

2.6.2 SHIELD GSM/GPRS

Si queremos ampliar las funcionalidades de nuestra plataforma Arduino, siempre podemos recurrir a una gran variedad de shields compatibles prácticamente con cualquiera de sus modelos. De este modo, podemos dotar al dispositivo de funciones adicionales dedicadas específicamente a ofrecer algún tipo de servicio concreto.

Un shield es un módulo de expansión en forma de placa impresa que se puede conectar a la parte superior de la placa Arduino para ampliar sus capacidades, permitiendo además ser apiladas unas encima de otras manteniendo un diseño modular.

Para hacer el prototipo del proyecto vamos a utilizar una shields SIM 900 GPRS/GSM, como la que podemos ver en la ilustración 14, configurada para ser controlada vía UART, utilizando los comandos AT. Para ver ejemplos del funcionamiento de esta shield podemos entrar en el portal de Prometec donde se expone paso a paso ejemplos de funcionamiento (Prometec, 2018).



Ilustración 14 Shields SIM 900 GPRS/GSM

Primero de todo deberemos insertar a la shield una tarjeta SIM activada (no importa la compañía telefónica).

Antes de conectar la shield al Arduino deberemos colocar dos jumpers en los siguientes pines de la shield para poder utilizar los pines 7 y 8 de nuestro Arduino para comunicarnos. Como bien se muestra en la ilustración 15.

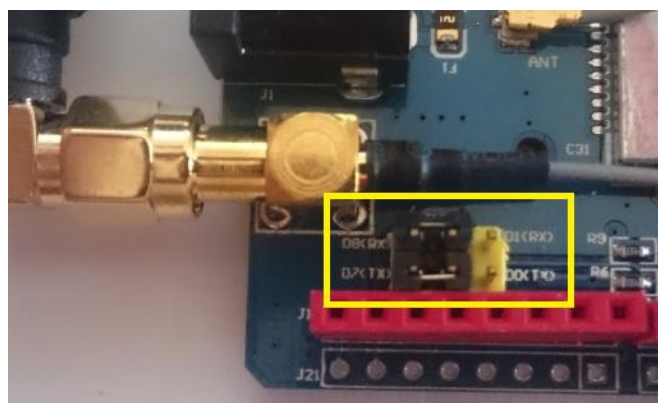


Ilustración 15 Colocación de los jumpers

Finalmente, solo faltará hacer el conexionado de la SIM900 al Arduino, es importante mencionar que requeriremos de una fuente de alimentación externa de unos 12V, ya que el consumo de la SIM900 es muy alto y si no utilizamos dicha alimentación externa, nos puede causar problemas a la hora de recibir y enviar SMS. El modo de conexionado con nuestra placa la podemos ver en la ilustración 16.

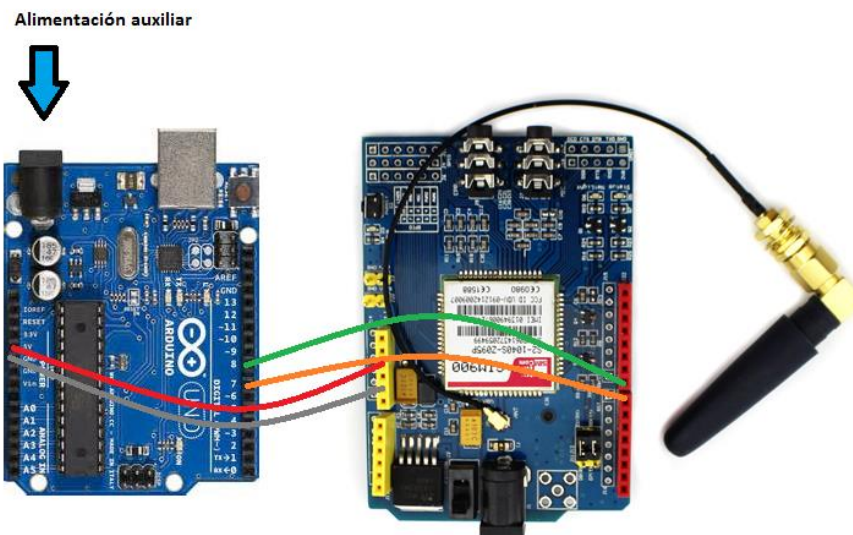


Ilustración 16 Conexionado SIM900 con Arduino

Una vez hecho el conexionado, podemos pasar a conectar el Arduino y la Shield a nuestro PC, si todo está correcto veremos que los Leds de la placa SIM900 se encenderán, como muestra la ilustración 17.

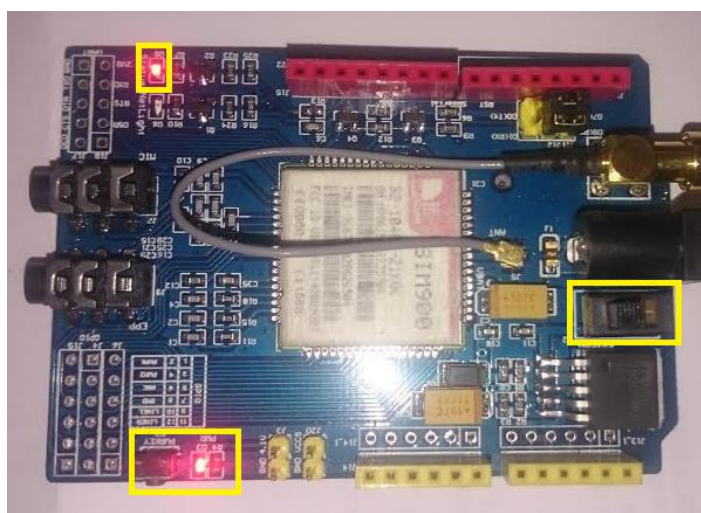


Ilustración 17 Encendido de los Leds en la Shield SIM 900

2.6.3 SHIELD RFID

Como módulo de lectura y escritura para el proyecto, utilizaremos el RC522, una pequeña placa que contiene un chip (MFRC522) como la que podemos ver en la ilustración 18, nos permitirá detectar nuestros tags RFID. En el portal de Prometec también podemos encontrar varios ejemplos del uso de este módulo, principalmente utilizado en control de accesos. (Prometec, 2016).



Ilustración 18 Shield RFID

El chip MFRC522, da soporte a la lectura y la escritura de los tags en diferentes condiciones y con control de errores, de una forma sencilla, aunque no puede detectar más de un tag a la vez, a diferencia de otros lectores más profesionales, también requiere que el tag este a un par de centímetros del receptor. En nuestro caso, como queremos hacer un pequeño prototipo de lector de tags, nos irá perfectamente.

En la ilustración 19, podemos ver el bus de pines del módulo y su conexión con el Arduino. Como podréis ver, únicamente hace falta conectar los pines de control SPI y la alimentación.

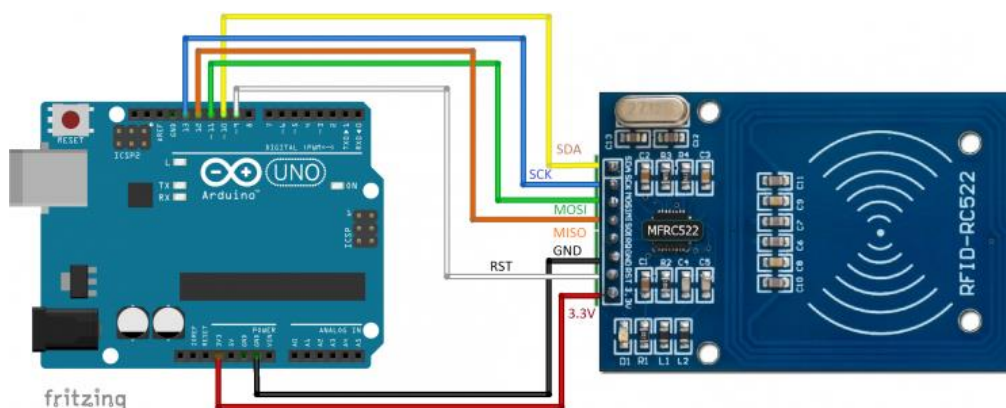


Ilustración 19 Conexión Shield RFID y Arduino

Una vez hechas las conexiones, podremos empezar a utilizar el módulo para leer el código de identificación de un tag.

CAPITULO III. DESARROLLO DEL PROTOTIPO

En este trabajo de fin de grado, vamos a realizar un pequeño prototipo de lector de etiquetas tag y el envío de información a un teléfono móvil vía SMS y registraremos la actividad del prototipo utilizando una web/APP. Con ello se pretende presentar una simulación de cómo funcionaría el SPK-Connect con una máquina de enfardado, demostrando que la idea de concepto es posible.

3.1 OBJETIVOS DEL PROTOTIPO

1. Ser capaces de detectar si hay un tag presente.
2. En caso de haber un tag presente, definir el tipo de bobina.
3. Contabilizar el número de ciclos que ha hecho la máquina.
4. Determinar el consumo por palet.
5. Bloquear la maquina en caso de no encontrar tag, salto de alarma o fin de bobina.
6. Solicitar vía SMS el estado de la máquina.
7. Enviar aviso en caso de emergencia.
8. Visualizar mediante una web el consumo de la máquina, localización, alarmas, etc.

3.2 DISECCIÓN DEL PROBLEMA.

Primero de todo vamos a realizar una partición del sistema para poder ver las diferentes partes involucradas y como se relacionan entre ellas.

Como podemos ver en la ilustración 20, el prototipo estaría formado por seis partes o módulos.

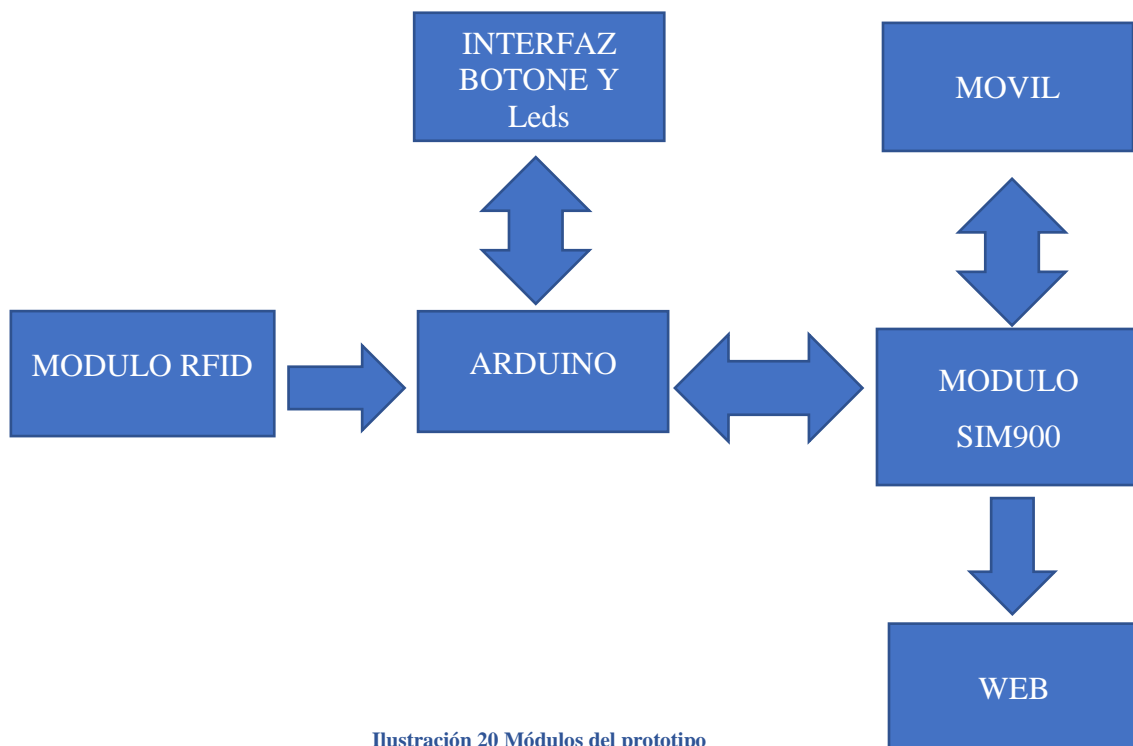


Ilustración 20 Módulos del prototipo

Para poder realizar este prototipo nos centraremos en los dos actuadores, el SIM 900 y el RFID. Realizaremos varias pruebas por separado para poder aprender correctamente su funcionamiento y seguidamente pasaremos a hacer la integración completa de todo el sistema.

3.2.1 LECTURA DEL TAG

Para el test del RFID, vamos a crear un sencillo programa que nos muestre por el puerto serial el código de identificación del tag, como podemos ver en la ilustración 21, el programa seguirá la siguiente función.

El programa deberá identificar si hay un tag que se pueda leer y seguidamente mostrar el código del tag por el puerto serial.

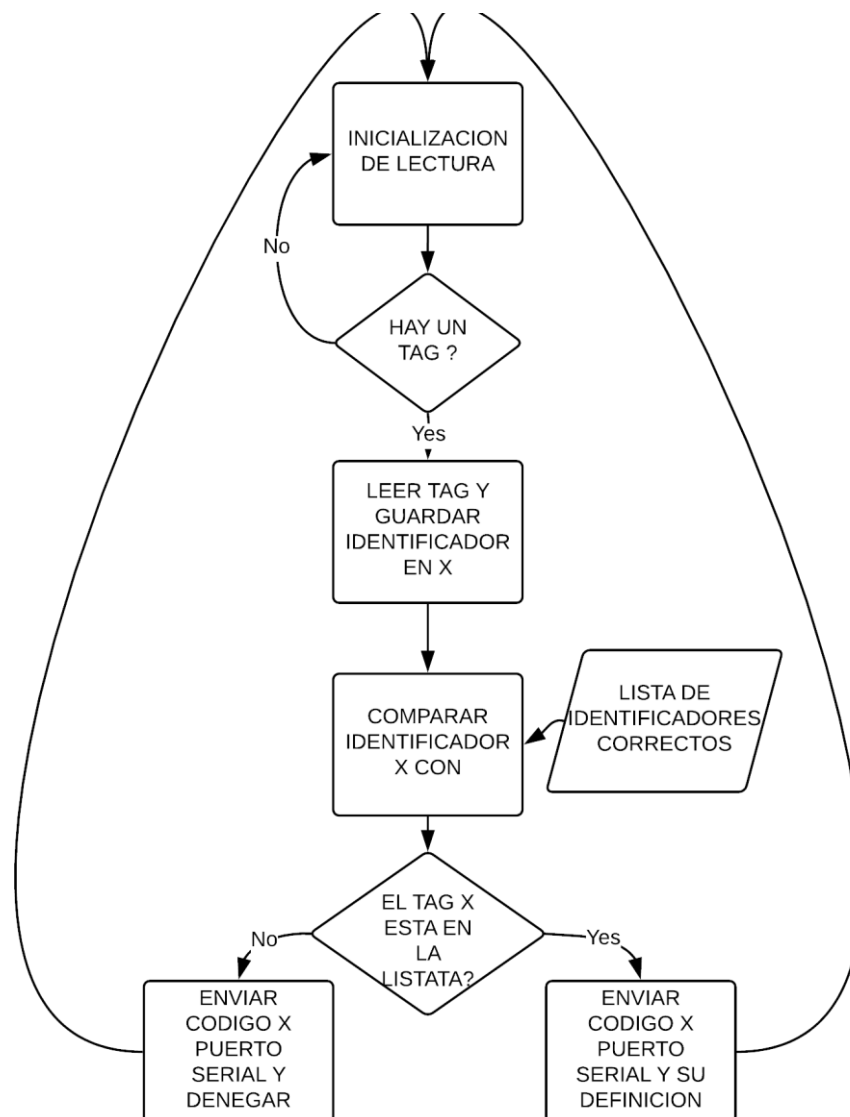


Ilustración 21 Diagrama de flujo RFID

3.2.2 CONTROL DEL ARDUINO POR SMS

Para el test del SIM900 vamos a crear un programa que nos permita encender un LED al enviar un SMS y nos devuelva otro SMS con el estado del LED. La ilustración 22 describe el funcionamiento del programa.

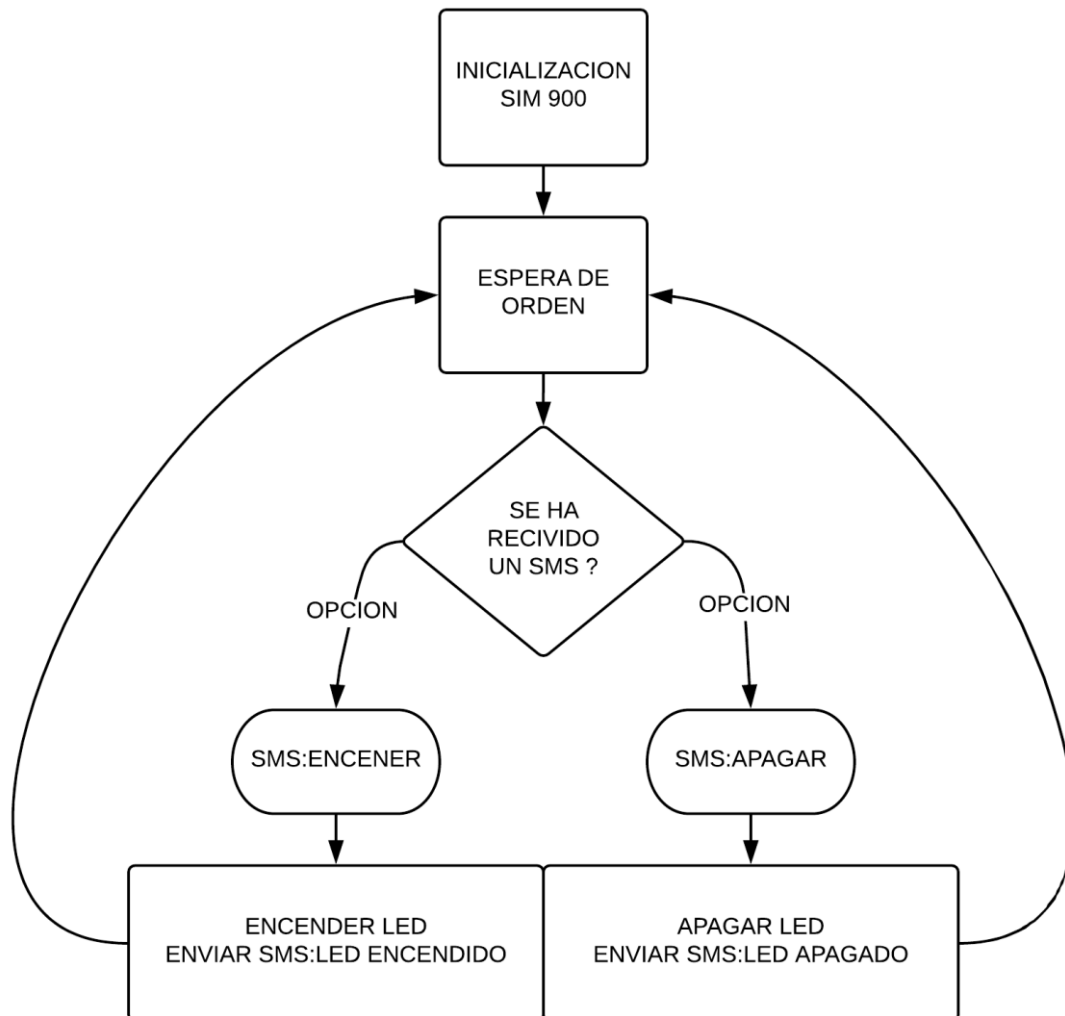


Ilustración 22 Diagrama de flujo SIM900

3.3 MONTAJE

3.3.1 MATERIAL

Una vez realizados los test y conozcamos el funcionamiento de todos los componentes vamos a realizar el montaje de todo el prototipo, el material del que vamos a disponer será:

- **2 pulsadores**

Utilizaremos pulsadores como actuadores.



EL BOTON VERDE SE
UTILIZARÁ PARA INICIALIZAR
LOS CICLOS DE ENFARDADO



EL BOTON ROJO SE UTILIZARÁ
COMO BOTON DE PARADA DE
EMERGENCIO Y/O ALARMA

- **3 Leds**

Se usarán Leds para dar información visual.

	INICIO DEL CICLO DE ENFARDADO
	RECEPCION DE UN SMS
	ALARMA O EMERGENCIA

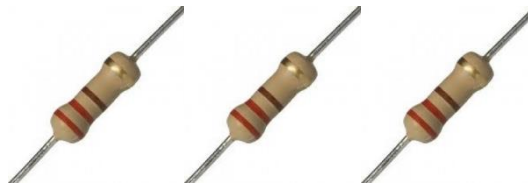
- **20 cables**

Se necesitarán para poder realizar las diferentes conexiones.



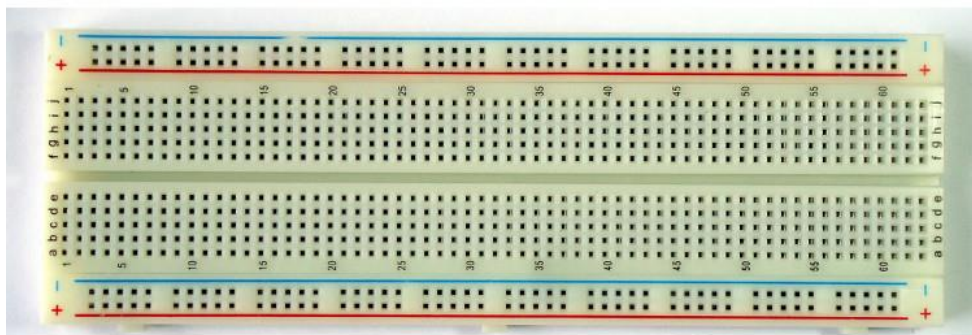
- **3 resistencias de 220 ohms**

Se utilizarán tres resistencias de 220 ohms para no quemar los Leds.



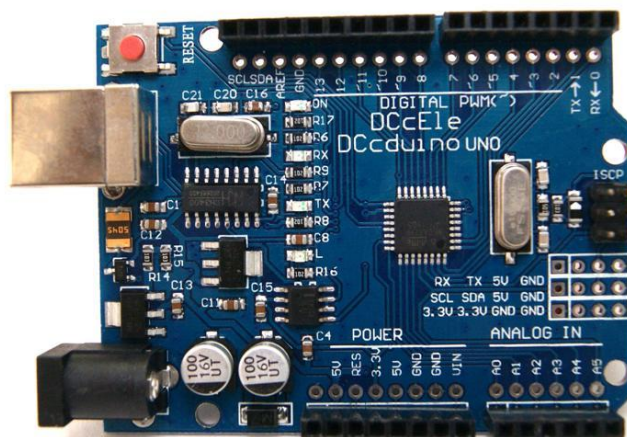
- **1 Protoboard**

La protoboard servirá como soporte para poder realizar todo el conexionado.



- **1 Arduino UNO R3**

El Arduino será el corazón del proyecto, el Arduino UNO R3 posee el número idóneo de entradas y salidas digitales para poder realizar el prototipo.



- **1 Shield SIM 900**

El SIM 900 es el módulo de comunicaciones móviles.



- **1 tarjeta SIM**

La tarjeta SIM es fundamental para poder enviar y recibir información ya que en esencia el Arduino actuara como si fuese un móvil.



- **1 Shield RFID**

El módulo RFID es la antena receptora para poder leer los códigos de los tags.



- **2 Tags**

Utilizaremos dos tags con códigos diferentes para poder asociar diferentes propiedades a cada Tag.



3.3.2 PRESUPUESTO

En la ilustración 23 podemos ver la tabla con el presupuesto del material necesario para poder llevar a cabo del prototipo.

2 pulsadores	0.20€
3 leds	1.50€
20 cables	2.00€
3 resistencias de 220 ohms	0.03€
1 protoboard	3.50€
1 Arduino UNO R3	4.50€
1 Shield SIM900	25.00€
1 tarjeta SIM	10.00€
1 Shield RFID	4.00€
2 tags	1.00€
Total	51.73€

Ilustración 23 Tabla de precios

3.3.2 CONEXIONADO

Entradas / Salidas	Pines Arduino
SDA - RFID	D10
SCK - RFID	D13
MOSI - RFID	D11
MISO - RFID	D12
GND - RFID	GND
RST - RFID	D9
3.3V - RFID	3.3V
D7 – SIM900	D7
D8 – SIM900	D8
5V – SIM900	5V
GND – SIM900	GND
LED AZUL	D2
BOTON ROJO	D5
LED ROJO	D4
LED BLANCO	D3
BOTON VERDE	D6

En la ilustración 24 podemos ver una tabla con las entradas de cada dispositivo y la conexión al Arduino, se ha segmentado la tabla con colores: azul para conexiones del RFID, naranja para las conexiones con del SIM900 y verde para la interfaz. En la ilustración 25 podemos ver el esquema de conexionado montado en la protoboard.

Ilustración 24 Tabla de conexiones

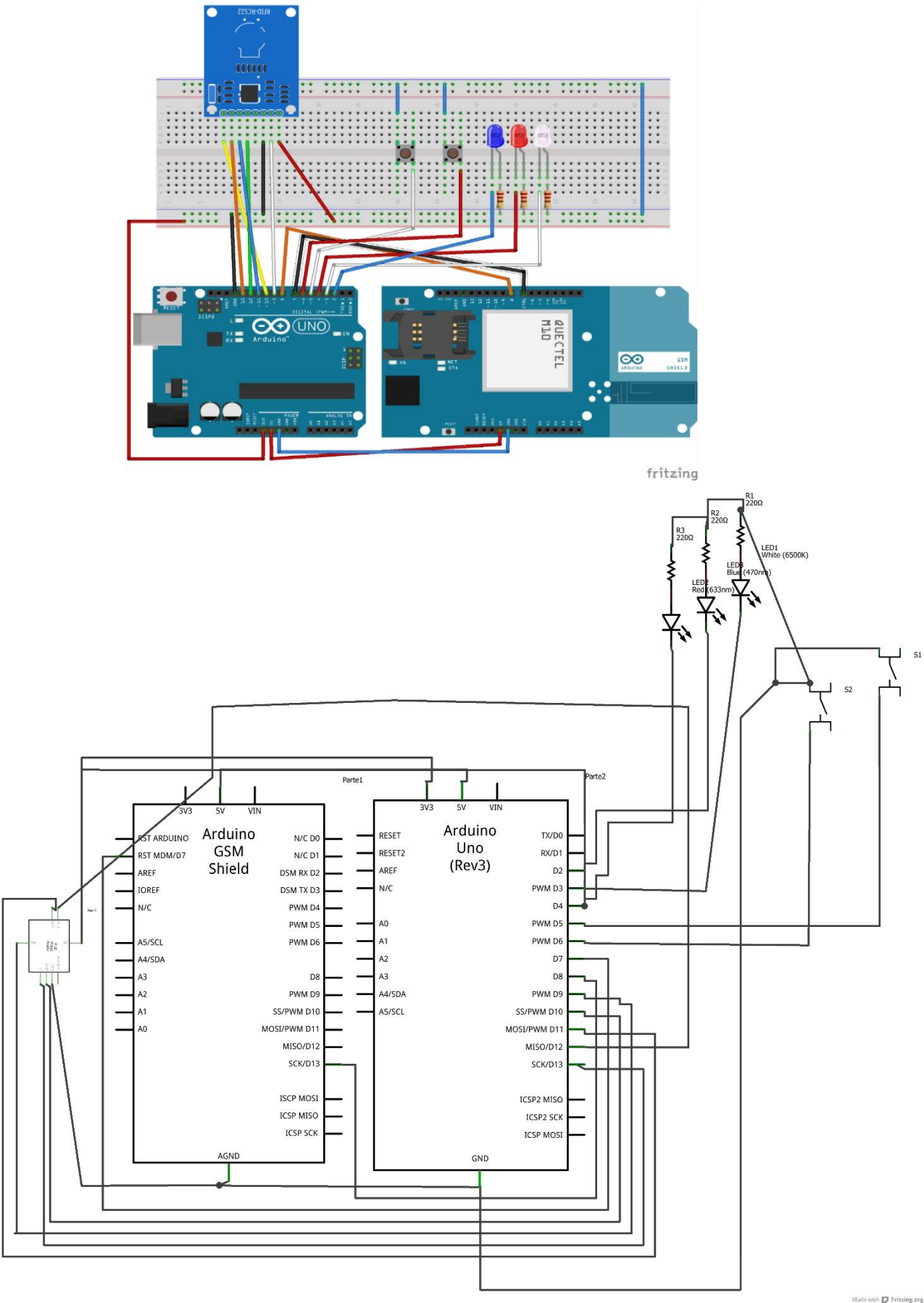


Ilustración 25 Esquema de conexiones

3.4 PRINCIPIO DE FUNCIONAMIENTO

El principio de funcionamiento es el siguiente:

Se inicializa el módulo de comunicación y se configura los parámetros de inicio junto con una petición para establecer las vueltas del programa de enfardado y el preestiro que se aplicará.

Al iniciarse la máquina entraremos en un estado de espera hasta que se active una alarma o se inicie un ciclo de enfardado.

Al iniciarse un ciclo se pueden dar dos casos:

- Caso 1, el Arduino no detecta o lee un tag que no está registrado en su base de datos con lo cual la maquina se bloquea automáticamente hasta que no se detecte un tag con un código correcto.
- Caso 2, el tag es correcto, se define la bobina a la cual corresponde el código y se prosigue a iniciar el ciclo de enfardado.

Una vez iniciado, se calcula el consumo de film que tendrá el ciclo de enfardado, este cálculo se realiza en base a los datos del programa de enfardo (número de vueltas y preestiro) y la información que facilita el tag de la bobina (peso, micraje y densidad del material) con lo cual obtenemos la siguiente fórmula:

$$Consumo = \frac{Hb \times 4000 \times Mc \times V \times d}{Pres} \frac{1}{10000}$$

Donde:

Hb: altura de la bobina =50cm

4000: perímetro de un europalet = 1200+800+1200+800mm

Mc: micraje de la bobina (de 15 a 35 micras)

V: número de vueltas que realizará la máquina

d: densidad del material LLDPE (polietileno de densidad lineal baja) = 0.916 g/cm³

Pres: preestiro que aplica la máquina gracias al hacer pasar el film por dos rodillos con velocidades diferentes, de esta manera para un presito de (250%) de 1m obtenemos 2,5m.

Ejemplo:

Queremos enfardar un europalet utilizando una bobina de 23micras aplicando 17 vueltas y un preestiro del 250%

$$\begin{aligned} Consumo &= \frac{Hb \times 4000 \times Mc \times V \times d}{Pres} \frac{1}{10000} \\ &= \frac{500 \times 4000 \times 23 \times 17 \times 0.916}{250} \frac{1}{10000} = 284,64g \end{aligned}$$

Una vez determinado el consumo que tendrá y sabiendo el peso de la bobina (16kg), podemos determinar el número de ciclos que le restan.

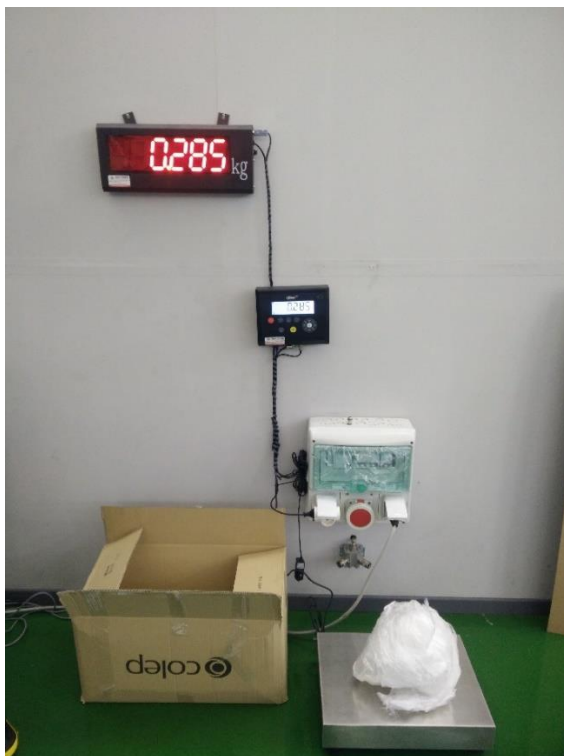


Ilustración 26 Peso del film

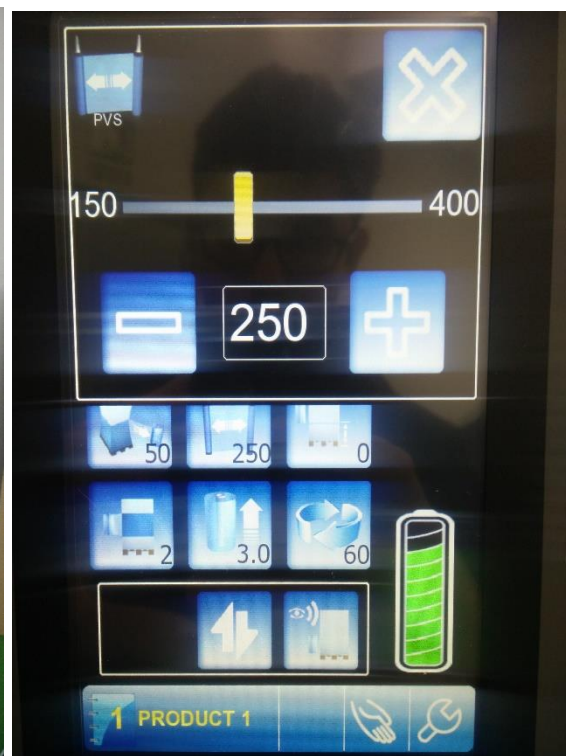


Ilustración 27 Panel de programación de una enfardadora

Como podemos observar en la ilustración 26, el peso de una bola de film de 23 micras, con un ciclo de envoltura de 17 vueltas y con 250% de preestiro, da un consumo de 285gramos por palet.

A continuación, vamos a definir las alarmas, interrupciones y código de colores.

INTERRUPCIONES	RESPUESTAS
Petición de estado de la máquina	SMS con información de la bobina, cuantos ciclos se han realizado y cuantos quedan por hacer
Alarma	Llamada y SMS alertando de la alarma Bloqueo de la máquina
Bobina a punto de terminar	SMS indicando los ciclos que quedan
Bobina finalizada	Llamada y SMS alertando de la alarma Bloqueo de la máquina

Código de colores

LED blanco fija	Ciclo en proceso
LED rojo fija	Alarma
LED rojo parpadeo	Alerta
LED azul parpadeo	Recepción de SMS
LED azul fija	Envío de SMS

En la ilustración 28 podemos ver el diagrama de flujo de la estructura del programa.

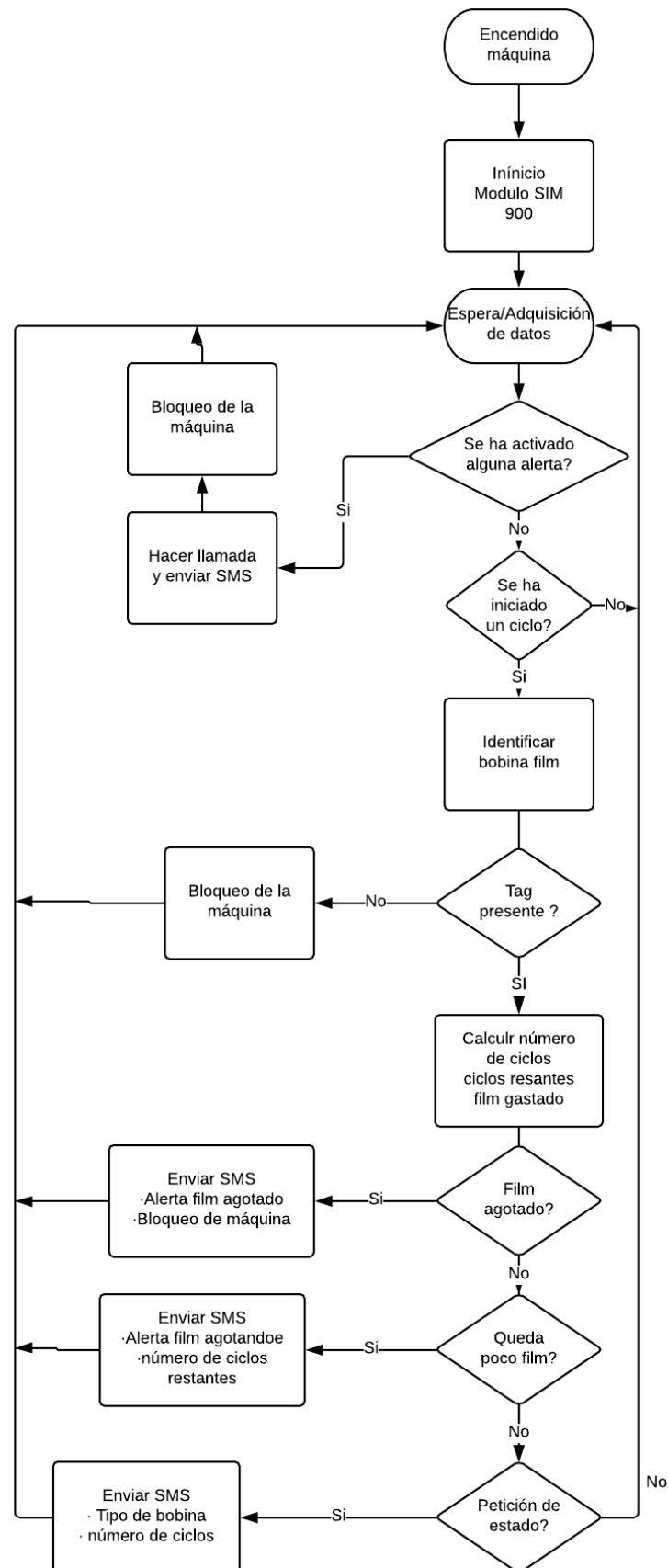


Ilustración 28 Diagrama de flujo del prototipo

3.5 CÓDIGOS

A continuación, se mostrarán las partes del código más relevantes junto con sus anotaciones.

El código completo lo podremos encontrar en los (Anexos II).

Declaraciones:

```
#include <BasicLinearAlgebra.h>
#include <SPI.h>
#include <MFRC522.h>
#include <SoftwareSerial.h>
#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522
MFRC522 mfrc522(SS_PIN, RST_PIN); ///Creamos el objeto para el RC522
SoftwareSerial SIM900(7,8); // Configura el puerto serie para el SIM900
//declaraciones
char incoming_char; //Variable que guarda los caracteres que envia el SIM900
String mensaje = "";
int LED = 2 ;// def LED Recepcion SMS
int Boton=5;//def Boton ALARMA
int AlarmaLED=4; //def LED ALARMA
int LEDciclo=3;//def LED CICLO
const int BOTONciclo=6;//def Boton CICLO
int val; //def valor de estado
int contador;//def contador de ciclos
bool estadoalarma = false; //def estado de alarma
byte ActualUID[4]; //almacenará el código del Tag leído
byte Bobina1[4]= {0xA5, 0xA7, 0x22, 0xC3} ; //código de bobina 1
byte Bobina2[4]= {0x13, 0x2E, 0x08, 0x85} ; //código de bobina 2
float consumo; //registro del consumo de film
float peso;//registro del peso de la bobina
float vueltas;//vueltas que da el palet
float micras;//micras del film
float densidadLLDPE=0.916;//constante densidad film
float largobobina=50;//constante altura bobina
float preestiro;//variable preestiro
bool fin_bobina=false;//estado de fin de bobina
bool lectura_bobina=false;//estado de lectura del tag
int alarma=0;//inicio estado alarma
```

Inicialización:

```

void inicializaSIM900()//rutia inicio SIM900
{
  SIM900.begin(19200); //Configura velocidad del puerto serie para el SIM900
  Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
  Serial.println("OK");
  delay (1000);
  SIM900.println("AT + CPIN = \"1995\""); //Comando AT para introducir el PIN de la tarjeta
  delay(25000); //Tiempo para que encuentre una RED
  Serial.println("PIN OK");
  SIM900.print("AT+CLIP=1\r"); // Activa la identificación de llamada
  delay(1000);
  SIM900.print("AT+CMGF=1\r"); //Configura el modo texto para enviar o recibir mensajes
  delay(1000);
  SIM900.print("AT+CNMI=2,2,0,0,0\r"); // Saca el contenido del SMS por el puerto serie del GF
  delay(1000);
  Serial.println("esperando ordendes...");
}

void datos_de_entrada()
{
  //Pedir datos del programa de enfardado
  Serial.println("Vueltas que dar al palet?");
  vueltas=leer();
  Serial.print(" ");
  Serial.print(vueltas);
  Serial.println(" ");
  Serial.println("Preestiro del film?");
  preestiro=leer();
  Serial.print(" ");
  Serial.print(preestiro);
  Serial.println(" ");
}

```

Inicio del ciclo:

```

void loop()
{
  val=digitalRead(BOTONciclo);//leer estado del boton
  if (val==LOW && fin_bobina==false && lectura_bobina==true)// si se inicia un ciclo...
  {
    digitalWrite(LEDciclo,HIGH);//encender LED ciclo
    contador=contador+1;//incrementar contador
    Serial.print("numero de ciclos: ");
    Serial.print(contador);//mostrar num ciclos
    Serial.println(" ");
    DB();// rutina de subir datos a la web
    Serial.print("gramos de film gastados: ");
    Serial.print(contador*consumo); //consumo acumulado
    Serial.println(" ");
    Serial.print("ciclos restantes: ");
    Serial.print((peso/consumo)-contador);// num Ciclos restantes
    Serial.println(" ");
    delay(2000);
    digitalWrite(LEDciclo,LOW);//fin del ciclo
  }
}

```



```

if(((peso/consumo)-contador)<=2)// en caso de que queden
{
    // pocos ciclos
    digitalWrite(AlarmaLED,HIGH);
    delay(1000);
    digitalWrite(AlarmaLED,LOW);// LUZ de alerta
    delay(1000);
    digitalWrite(AlarmaLED,HIGH);
    delay(1000);
    digitalWrite(AlarmaLED,LOW);
    delay(1000);
    digitalWrite(AlarmaLED,HIGH);
    delay(1000);
    digitalWrite(AlarmaLED,LOW);
    Alerta_film();// envio de SMS

    if ((peso/consumo)==contador)// film AGOTADO
    {
        digitalWrite(AlarmaLED,HIGH);
        Alarma_film();//envio SMS
        llamar();//llamada emergencia
        fin_bobina=true;// estado bobina acabado
    }
}

```

Identificador de entrada de petición:

```

if (incoming_char == '?')//SI envia una SMS con ?...
{
    digitalWrite(LED,HIGH);
    delay(500);
    digitalWrite(LED,LOW);
    delay(500);
    digitalWrite(LED,HIGH);//indica que se ha recibido un sms
    delay(500);
    digitalWrite(LED,LOW);
    delay(500);
    digitalWrite(LED,HIGH);
    delay(500);
    digitalWrite(LED,LOW);

    if(compareArray(ActualUID,Bobina1))
    {
        mensaje_smsBobina1();// manda un sms con el tipo de bobina
    }//subrutina envio tipo de bobina 1
    if(compareArray(ActualUID,Bobina2))
    {
        mensaje_smsBobina2();// manda un sms con el tipo de bobina
    }//subrutina envio tipo de bobina 2
}

```

Envío de información:

```

void mensaje_smsBobinal()//rutina envio info bobina
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\\");
    delay(1000); //Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Bobina 23 my 16kg prestiro");// Texto del SMS
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();|
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
    sms_ciclo();//rutina envio info ciclos
}

void num_ciclos()//Rutina SMS num ciclos
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\\");
    delay(1000); //Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Numero de ciclos:");// Texto del SMS
    delay(100);
    SIM900.println(contador);// Texto del SMS
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");|
    incoming_char == '0';
}

```

En caso de que la bobina no pertenezca a Control Pack:

```

void bobina_no_encontrada()
{ //Rutina que bloquea la máquina, bobina no valida
    Serial.println("BOBINA NO VALIDA");
    digitalWrite(AlarmaLED,HIGH);
    lectura_bobina==true;// bloqueo de la máquina
}

```

Llamada:

```

void llamar()//rutina llamada
{
    SIM900.println ("ATD 628565023;") ; // Número de telefono.
    delay ( 100 ) ;
    SIM900.println () ;
    delay ( 30000 ) ;           // Esperar 30 segundos...
    SIM900 . println ( "ATH" ) ; // Colgar
    Serial.println("Llamada finalizada");
    digitalWrite(AlarmaLED,LOW);
}

```

Identificación de bobina:

```

void infoBOBINA()//rutina info bobina
{
    if(mfrc522.PICC_IsNewCardPresent())
    {
        //seleccionamos una tarjeta
        if (mfrc522.PICC_ReadCardSerial())
        {
            // Enviamos serialamente su UID
            Serial.print(F("Card UID:"));
            for (byte i = 0; i < mfrc522.uid.size; i++)
            {
                Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
                Serial.print(mfrc522.uid.uidByte[i], HEX);
                ActualUID[i]=mfrc522.uid.uidByte[i];
            }
            Serial.print("      ");
            //comparamos los UID para determinar si es uno de  tras bobinas
            if(compareArray(ActualUID,Bobina1)){// bobina1
                Serial.println("Bobina de control Pack");
                Serial.println("Bobina 23 my 16kg prestiro");
                peso=16000;//def parametros bobina1
                micras=23;
                lectura_bobina=true;
                mfrc522.PICC_HaltA();
            }
            if(compareArray(ActualUID,Bobina2)){// bobina2
                Serial.println("Bobina de control Pack");
                Serial.println("Bobina 30 my 16kg prestiro");
                peso=16000;//def parametros bobina2
                micras=30;
                lectura_bobina=true;
                mfrc522.PICC_HaltA();
            }
            }
        }
        // Terminamos la lectura de la tarjeta tarjeta actual
        mfrc522.PICC_HaltA();
    }
}

```

Actualizar Base de Datos:

```

void DB()
{
  //rutina envio de datos a la web
  consumo=(largobobina*4000);
  consumo=(consumo*micras*densidadLLDPE);
  consumo=(consumo/preestiro);
  consumo=(consumo*vuelatas);
  consumo=(consumo/1000);
  //calculo del consumo de film

  inicializar();//Rutina para escribir en la web
  subir_consumo();//Rutina envío de consumo
  inicializar();//Rutina para escribir en la web
  subir_ciclos(); //Rutina envío de ciclos
  Serial.println(consumo);
  Serial.println(micras);
  Serial.println(preestiro);
  Serial.println(vuelatas);
  Serial.println(peso);
  //visualizar datos puerto serial
}

void inicializar()
{
  //Rutina para escribir en la web
  SIM900.println("AT+CIPSHUT");
  delay(1000);//cerrar posible conexion
  SIM900.println("AT+CIPMUX=0");
  delay(2000);//abrir conexion internet
  SIM900.println("AT+CGATT=1");
  delay(2000);//activar modo GPRS
  SIM900.println("AT+CSTT=\"internet\",\"orange\",\"orange\");
  delay(2000);//APN,username,pass
  SIM900.println("AT+CIICR");
  delay(2000);// establecer nuestra direccion IP
  SIM900.println("AT+CIFSR");
  delay(2000);
  SIM900.println("AT+CIPSTART=\"TCP\",\"184.106.153.149\",\"80\");
  delay(2000);// establecer direccion donde queremos ir
  SIM900.println("AT+CIPSEND=50");
  delay(2000);//establecer modo enviar dato
}

```

```
void subir_consumo()
{
    SIM900.print("GET /update?key=ARAV5UUQGK1JTYGH&field1=");
    delay(2000); //direccion donde colocar el valor
    SIM900.println(consumo);
    delay(2000); //dato que queremos enviar
    SIM900.println("#026"); //fin del envio
}

void subir_ciclos()
{
    SIM900.print("GET /update?key=ARAV5UUQGK1JTYGH&field3=");
    delay(2000); //direccion donde colocar el valor
    SIM900.println((peso/consumo)-contador);
    delay(2000); //dato que queremos enviar
    SIM900.println("#026"); //fin del envio
}
```

3.6 SIMULACIÓN

A continuación, vamos a realizar una simulación del funcionamiento real de una máquina automática de enfardado, como la que podemos ver en la ilustración 29, basándonos en el prototipo y apoyándonos en el diagrama de flujo que hemos visto anteriormente, ilustración 28.



Ilustración 29 Envolvedora de palets ROTOTECH 3000

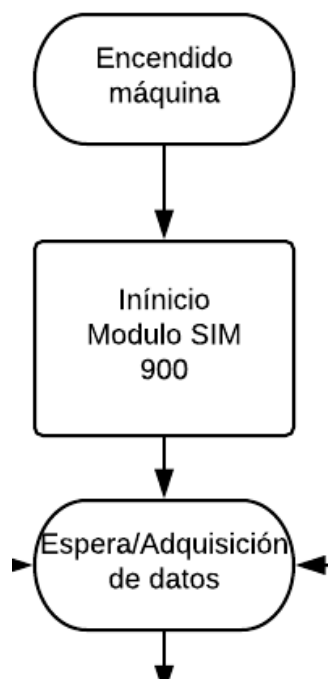
Inicialmente la máquina se iniciaría, se programaría la receta de enfardado para la línea de palets y quedaría en espera de recibir palets.

En el prototipo, hemos simulado esta etapa de la siguiente manera.

```
esperando ordendes...
Vueltas que dar al palet?
 17.00
Preestiro del film?
 250.00

Card UID: 13 2E 08 85      Bobina de control Pack
Bobina 30 my 16kg preestiro
```

Ilustración 30 Captura pantalla monitor Serial



Las variables que determinan el consumo de film en el palet son las que obtenemos a través del programa de enfardado (número de vueltas y preestiro), leemos el tag de la bobina para saber si es aceptable y para poder saber las características de la bobina (micraje).

La máquina al identificar el tipo de bobina que lleva en el carro y determinar la acción siguiente. Nos podemos encontrar con dos casos.

Caso 1: La máquina detecta el Tag, determina la clase de bobina con la que está trabajando y procede a envolver, mientras guarda un registro de los ciclos de enfardado etc.

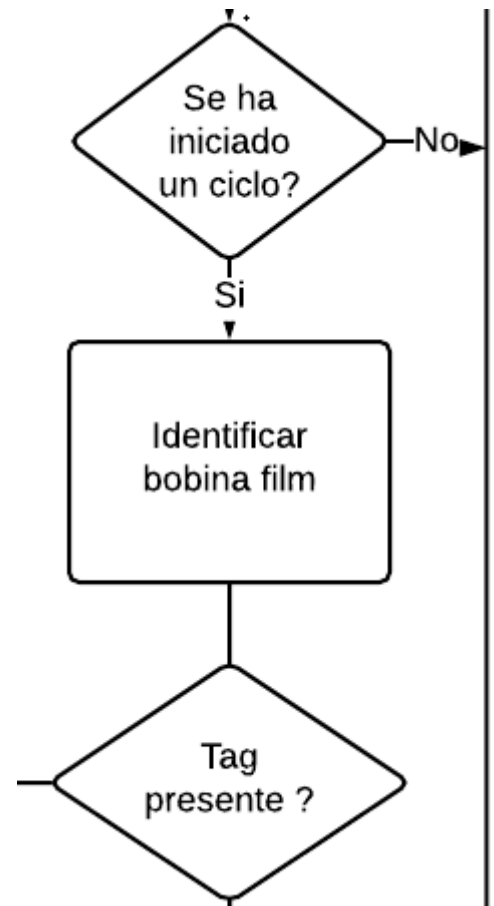
```
Card UID: 13 2E 08 85      Bobina de control Pack  
Bobina 23 my 750 gramos preestiro
```

Ilustración 31 Captura pantalla monitor Serial

En nuestro prototipo lo hemos simulando utilizando un botón verde para indicar la orden de inicio del ciclo y una luz LED blanca a modo de funcionamiento del ciclo.



Ilustración 32 Encendido LED blanco al pulsar botón verde



Caso 2: No se detecta el tag, con lo cual nos indica que o no hay una bobina en el carro o la bobina del carro no pertenece a Control Pack, con lo cual la máquina quedaría bloqueada, se encendería un LED rojo y procedería al aviso.



Ilustración 33 Encendido LED rojo al pulsar botón verde y no detectar tag monitor Serial

```

-----
BOBINA NO VALIDA
BOBINA NO VALIDA
BOBINA NO VALIDA
BOBINA NO VALIDA
BOBINA NO VALIDA
BOBINA NO VALIDA
-----

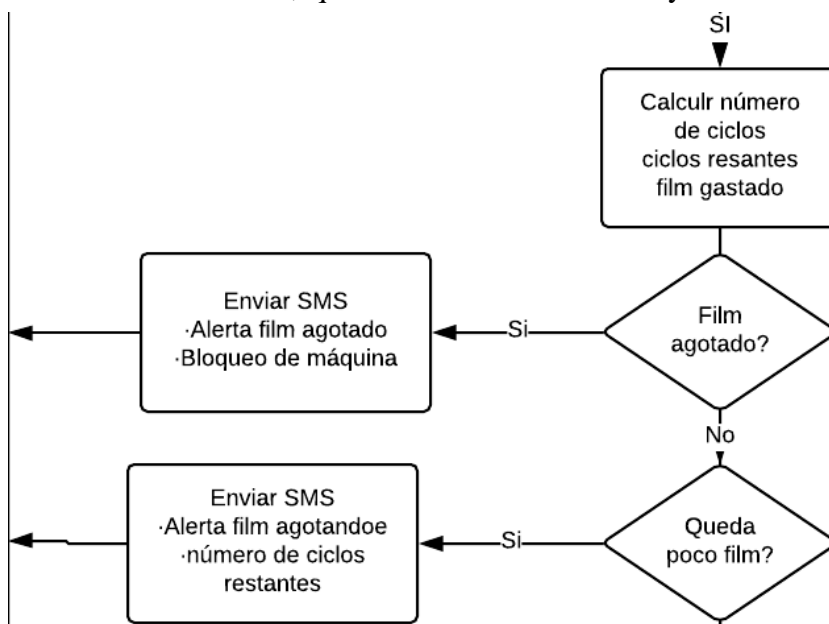
```

Ilustración 34 Captura pantalla

Durante el funcionamiento normal de la máquina, se irán guardando los datos de cada ciclo de enfardado, cuantos ciclos se han hecho, que consumo se ha tenido y cuantos ciclos le quedan a la bobina.

Mediante la plataforma Web ThingSpeak.com (ThingSpeak, 2018), que nos permite conectarnos con nuestro Arduino podremos enviar datos.

Hemos creado las siguientes tablas a modo de interfaz de usuario



para poder disponer de un histórico de la máquina y ver su evolución en tiempo real desde cualquier lugar.

La gráfica de la ilustración 35, nos muestra el histórico de consumo de film, en este caso, podríamos deducir que hay una variación de formato de palet o cambio de programa.

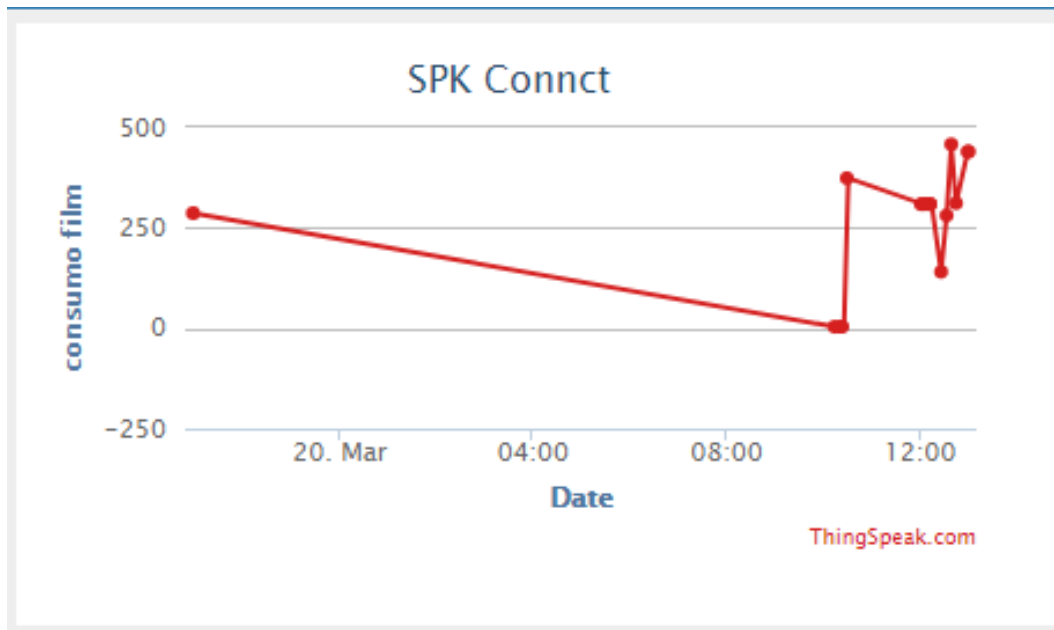


Ilustración 35 Consumo de film

A modo de registrar una alarma, se envía un 1 lógico indicando el problema. Para esta simulación únicamente se contempla una para de emergencia, pero se puede extrapolar a todas las alarmas de la máquina real. En la gráfica de la ilustración 36 podemos ver la activación de dos alarmas.

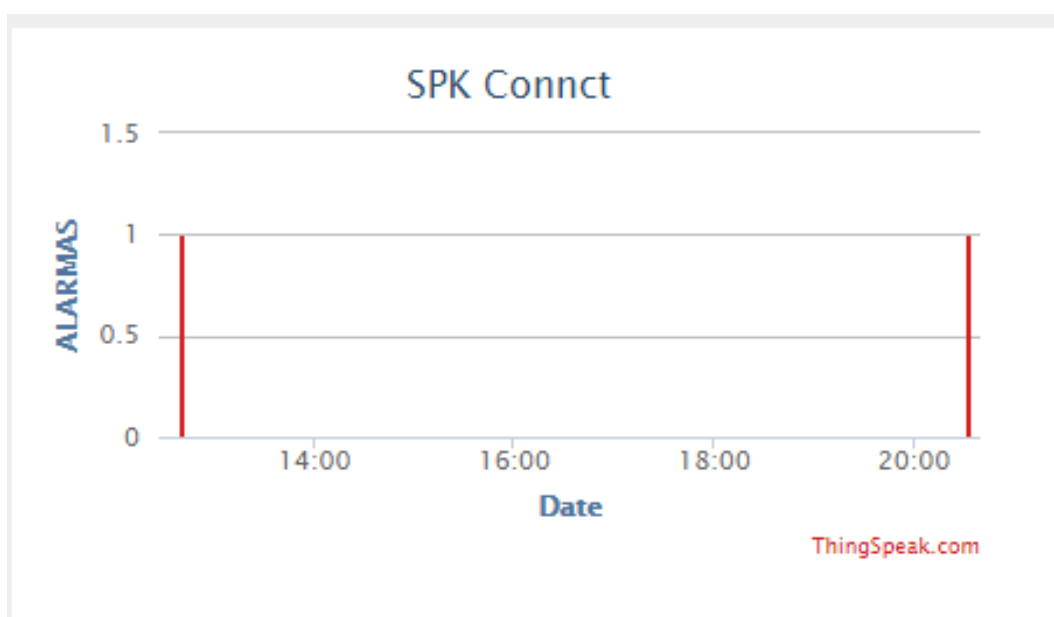


Ilustración 36 Activación de alarmas

Poder situar la máquina en el mapa es muy importante a la hora de gestionar reparaciones u organizar una visita para mostrar la máquina a un futuro cliente. La ilustración 37 es una ventana en la cual puedes indicar las coordenadas GPS de la ubicación de la máquina y te permite trazar un camino hasta ella.



Ilustración 37 Localización de la máquina

En base al consumo de film que tiene la máquina y sabiendo el peso de la bobina (16 kg), Podemos saber cuántos ciclos le va a durar. Este ejemplo muy sencillo servirá a la hora de anticipar la demanda del cliente, ya que sabiendo el número de bobinas que se le ha vendido y sabiendo las que ha consumido, podemos saber el remanente que tiene y hacer la gestión para ofrecerle más. De esta manera se puede planificar la producción a la demanda. La ilustración 38 sería un ejemplo.

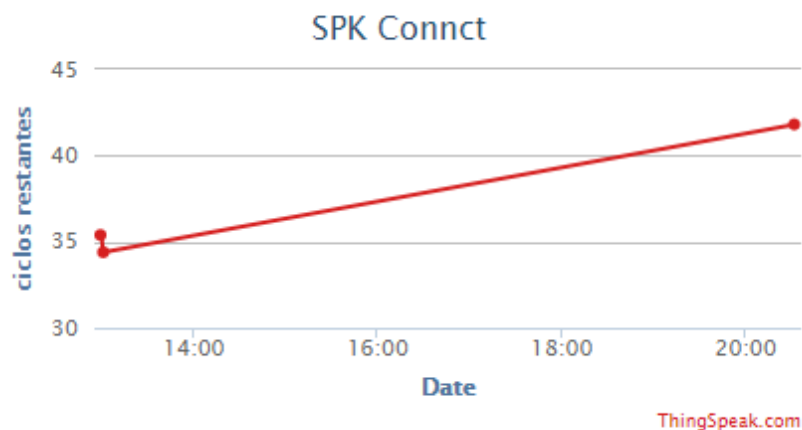


Ilustración 38 Ciclos de la máquina

En el caso de que a la bobina le queden pocos ciclos de enfardado que ofrecer, se procederá a dar una alerta indicando el número de ciclo que le restan a la bobina.

Una vez la bobina haya finalizado, se procederá a enviar una alarma al técnico,

Se ha realizado la siguiente simulación para reflejar estos dos estados.

Como se puede apreciar en la ilustración 39, la máquina nos ha ido avisando ciclo a ciclo que se estaba acabando el film de la bobina, también da un aviso luminoso con el LED rojo parpadeando, hasta que finalmente se ha agotado.

Una vez que la bobina se haya acabado, el LED rojo quedará encendido permanentemente y se procederá a hacer una llamada al técnico para que realice el cambio de bobina. Como se ve en la ilustración 41.



```
ciblos restantes: 0
Enviando SMS...
SMS enviado
Enviando SMS...
SMS enviado
Llamada finalizada
```

Ilustración 40 Captura pantalla monitor Serial

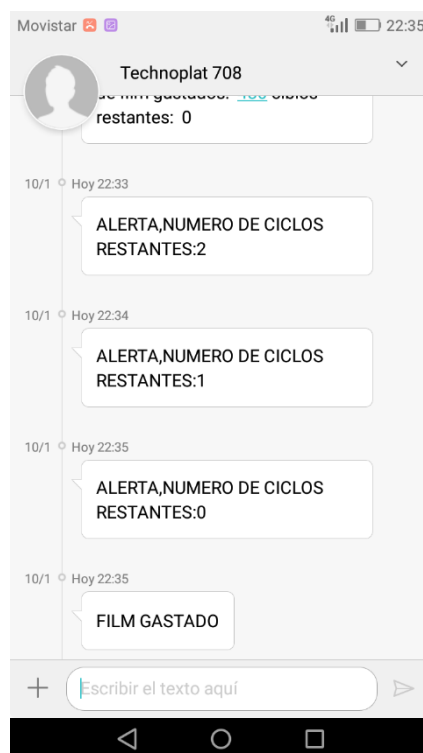


Ilustración 39 Alerta film gastado

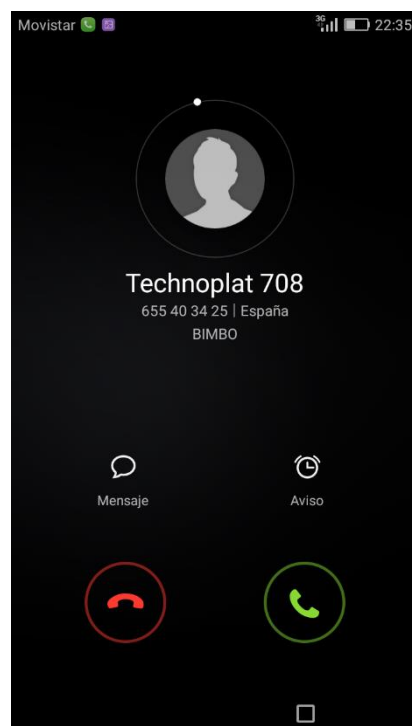


Ilustración 41 Llamada de la Máquina

Durante el funcionamiento de la máquina podemos hacer una solicitud de información del estado de la misma. Es decir, pedir telemáticamente vía SMS, saber con qué tipo de bobina está trabajando la máquina y los datos de consumo.

Para ello únicamente tendremos que enviar un SMS a la máquina con el símbolo “?”, tal y como se indica en la ilustración 43, la máquina recibirá la petición de información y nos devolverá un SMS con los datos pertinentes. A modo de información visual, cada vez que la máquina reciba una petición, parpadeara un LED azul indicando que ha recibido la comanda.

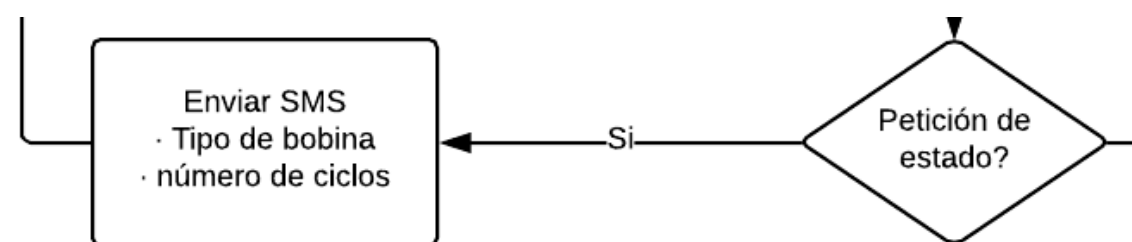


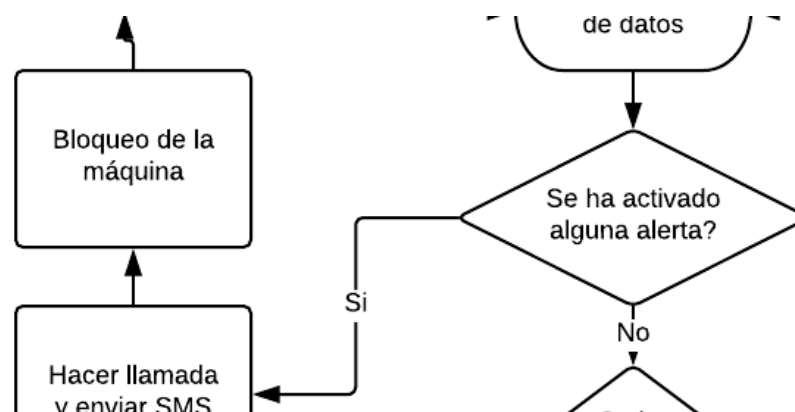
Ilustración 43 SMS Máquina



Ilustración 42 LED azul encendido al recibir una petición

Por último, un sistema de teleasistencia como este también tiene que tener en cuenta la seguridad, por ello en caso de una parada de emergencia, nos enviara una SMS indicando el tipo de Alarma y hará una llamada al técnico a modo de alarma a distancia.

Para poder simular dicha alarma, hemos incluido un botón rojo a modo de parada de emergencia, dicho botón simulara una de las tantas alarmas que pueden surgir en la máquina, como por ejemplo la rotura del film o la colisión del brazo rotatorio.



Una vez surgida la alarma, la máquina se detiene automáticamente y se encenderá el LED rojo, como hemo mencionado anteriormente, también enviará el SMS indicando que tipo de alarma como podemos ver en la ilustración 44-45, y la consiguiente llamada a modo de aviso al operario.

La máquina permanecerá bloqueada hasta que no se vuelva a rearmar, por ello en nuestra simulación se realizará un “reset” del Arduino.



Ilustración 44 LED rojo al pulsarse botón alarma

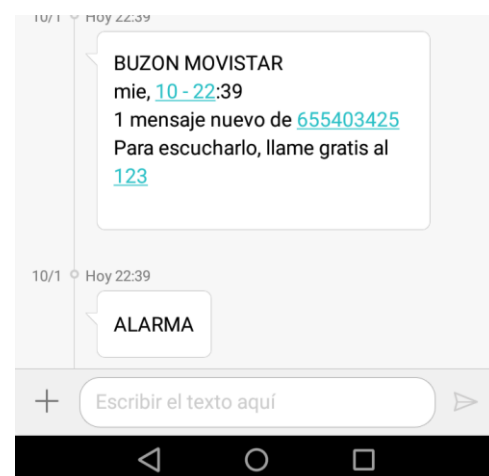


Ilustración 45 SMS Máquina

CAPITULO IV. WRAPSTER

Actualmente el proyecto del SPK-Connect se está llevando a cabo dentro de la empresa como un proyecto de I+D el cual ya ha recibido una subvención del CDTI (Centro para el Desarrollo Tecnológico Industrial).

La máquina que integrará una versión temprana del SPK-Connect será la WRAPSTER como la de la ilustración 48, la fusión entre una enfardadora de palets automática como la ilustración 46 y una flejadora de palets, ilustración 47, de mano de la empresa Reisopack, empresa colaboradora.

Actualmente el proyecto está en fase de pruebas.

Tareas completadas:

- Definición de alarmas
- Montaje e instalación de las máquinas
- Montaje e instalación armario conexiones y PLC
- Programa PLC
- Montaje Modulo SIM
- Panel de control HMI

Tareas por realizar:

- Servidor propio
- Interfaz web
- Comunicación con servidor
- Detección de tag RFID
- Instalación célula de carga

En este último capítulo hablaremos sobre la puesta en marcha de la máquina, así como de mi implicación personal en el proyecto y las partes más destacas de este.

Trabajos realizados en la WRAPSTER:

Definir secuencia del ciclo de la máquina y alarmas.



Ilustración 46 Technoplat



Ilustración 47 Flejadora 2200



Ilustración 48 WRAPSTER

“Estación automática de flejado y envoltura de palets modelo Wrapster. Con esta solución el operario solo debe posicionar el palet encima de la plataforma con toro o transpaleta y pulsar un botón para que Wrapster deje el palet perfectamente estabilizado” (CONTROLPACK, 2018)

La combinación de una envolvedora de palets y una flejadora de arco automática, nos da como resultado la WRAPSTER.

4.1 LISTADO DE ALARMAS DE LA WRAPSTER

A continuación, veremos en la ilustración 49, una tabla con las alarmas que se han definido para este sistema.

También veremos algunos actuadores para poder hacer el control de cambio de bobina como se puede apreciar en la ilustración 51.

ALARMAS WRAPSTER		
Identificador	Descripción	Dirección PLC Omron
E01	Pulsador E-stop activo	W_Bit 15.00
E02	Fallo enhebrado	W_Bit 15.01
E03	LS1 activo	W_Bit 15.02
E04	LS2 Activo	W_Bit 15.03
E05	Cabezal fuera de posición	W_Bit 15.04
E06	By-pass activo	W_Bit 15.05
E07	Fallo térmico	W_Bit 15.06
E08	Fallo inverter	W_Bit 15.07
E09	Tope salida lanza	W_Bit 15.08
E10	Tope retorno lanza	W_Bit 15.09
E11	Fin bobina fleje	W_Bit 16.00
E12	Fallo en mesa giratoria	W_Bit 16.01
E13	Puerta cabezal abierta	W_Bit 17.00
E14	Seguridades activas	W_Bit 17.01
E15	Puerta vallado abierta	W_Bit 17.02
E16	Barrera de seguridad entrada activa	W_Bit 17.03
E17	FTC. Transpaleta ocupada	W_Bit 17.04
E18	Mesa envolvente no en fase	W_Bit 17.09
E19	Encoder 1 no cuenta pulsos	W_Bit 17.10
E20	Encoder 2 no cuenta pulsos	W_Bit 17.11
E21	Paro activo	W_Bit 17.12
E22	Descompensación de posición, realizar reset de posición	W_Bit 17.13
E23	Realizar reset máquina	W_Bit 17.14
DATOS PRODUCCIÓN WRAPSTER		
Identificador	Descripción	
P01	Palets procesados	INT_ H180
P02	Cambio de bobina	INT_ H181
P03	Vueltas mesa de giro	INT_ H182

Ilustración 49 Tabla de alarmas de la WRAPSTER y entrada de datos

La lectura de las alarmas se obtiene a partir de las salidas digitales de los PLC de la envolvedora y la flejadora, igual que los datos de producción.



Ilustración 50 Panel de control de la WRAPSTER

Como podemos ver en la ilustración 50, el HMI de la WRAPSTER y la envolvedora de palets, la pantalla muestra que ha saltado una alarma, en este caso se ha pulsado la seta de emergencia.

Como hemos mencionado anteriormente, en esta versión de la WRAPSTER hemos contemplado un sistema de conteo de cambio de bobina basado en una fotocélula y una superficie reflectante. Este sistema se ha implementado a falta de instalar la antena RFID. Por tanto, la máquina medirá el consumo a base de cambios de bobinas y palets enfardados. Por otra parte, a modo de control de acceso, la máquina no funcionará en caso de que detecte que no hay bobina. Este sistema lo podemos ver en la ilustración 50.

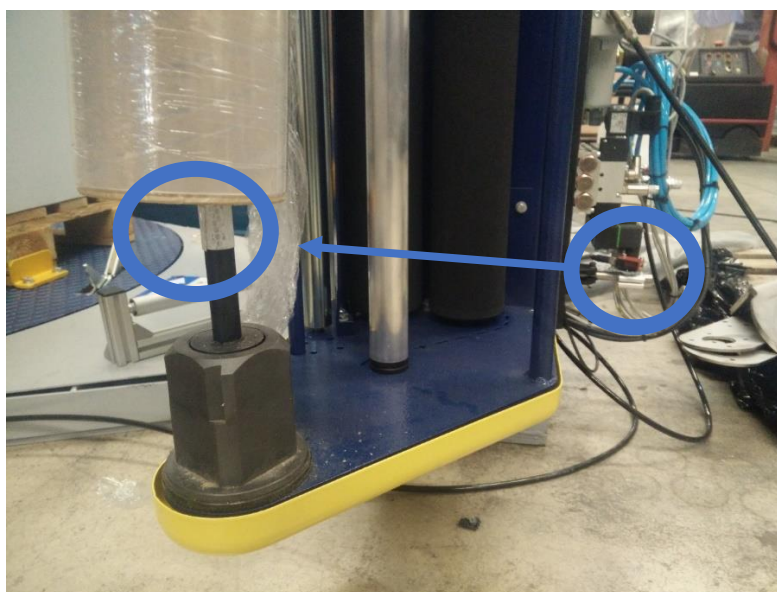


Ilustración 51 Sistema para contar cambios de bobina

En la ilustración 52, podemos ver el HMI de la WRAPSTER, concretamente el panel que nos da la información de consumo de la máquina.

3 bobinas de 16 kg de film

63 palets enfardados

557 vueltas

$$\frac{3 \times 16}{63} = 0.76kg \text{ por palet}$$

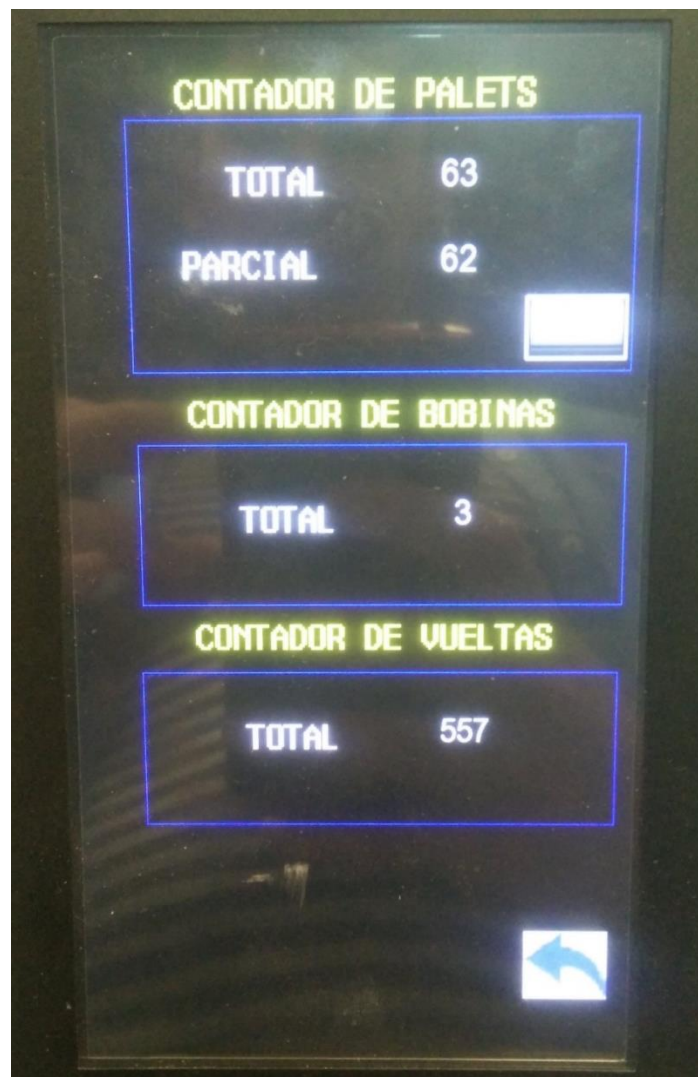


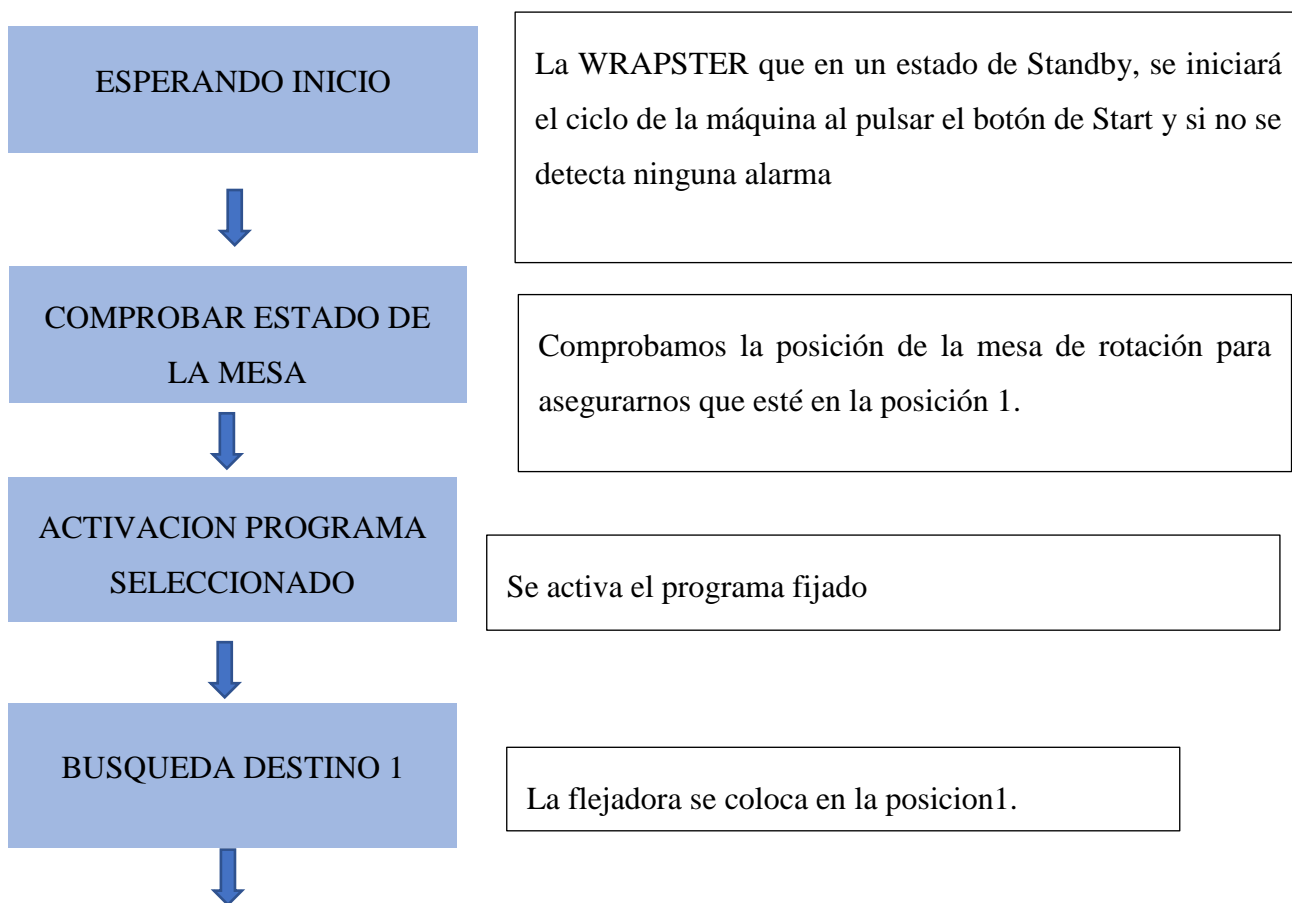
Ilustración 52 Datos de la WRAPSTER

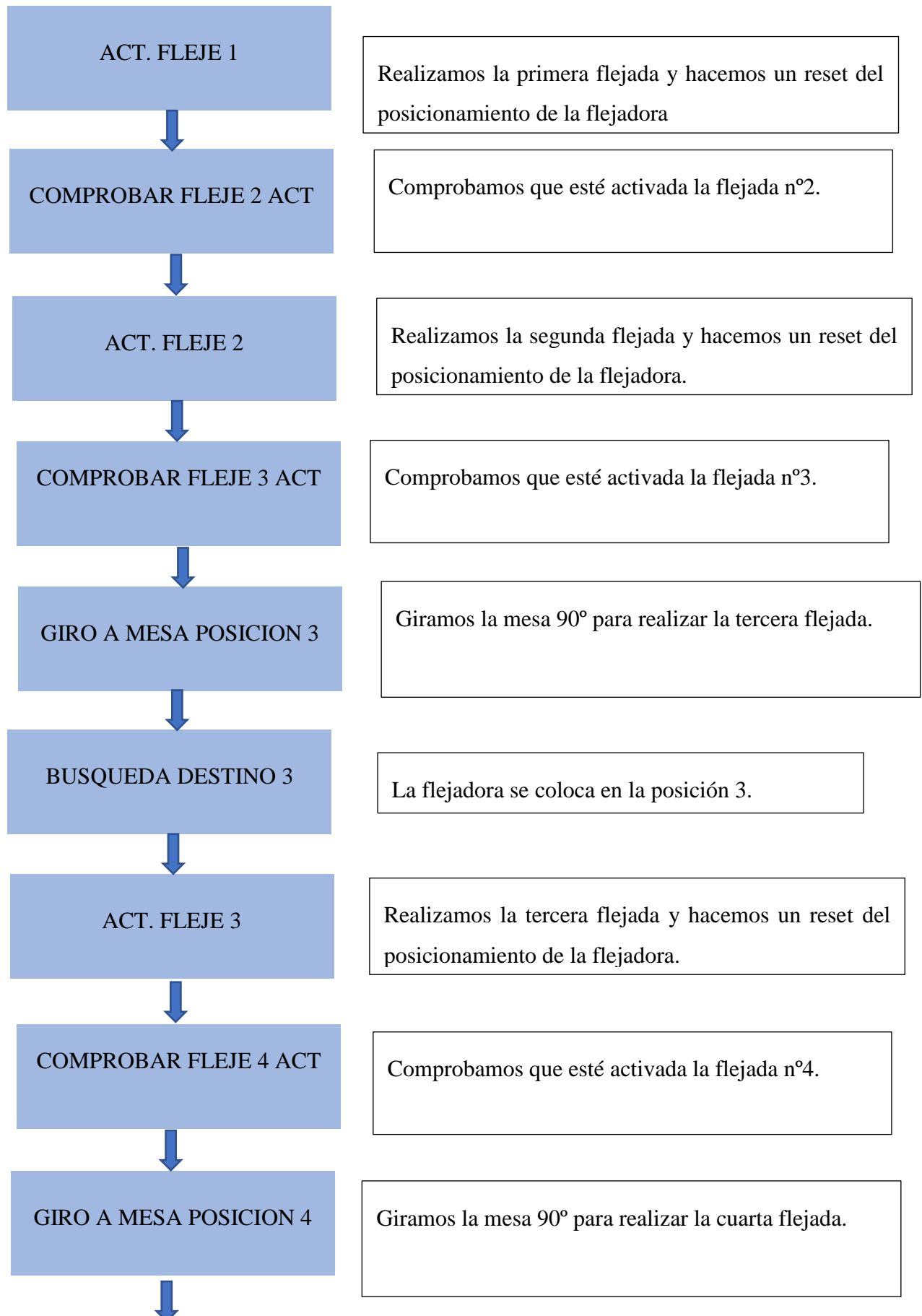
4.2 DIAGRAMA DE SECUENCIA DE LA WRAPSTER

Este es el diagrama de secuencia en el cual se ha basado el programa del PLC para poder hacer la WRAPSTER.

Se puede observar que prácticamente todo el proceso se basa únicamente en la flejadora, eso es debido a que la flejadora en si no tiene un programa predeterminado para hacer la flejada, sino que hemos adaptado la flejadora a la enfardadora. Actualmente podemos elegir la posición y el número de flejadas a realizar. Al tener la flejadora encima de unos railes, podemos determinar su posición, por otro lado, el palet al estar encima de la mesa giratoria de la enfardadora, también podemos elegir si queremos flejar en cruz o en paralelo.

Finalmente, una vez acabado el proceso de flejar, activamos el sistema de enfardado, previamente programado. En la ilustración 53 podemos observar los dos paneles para programar el ciclo de la máquina.





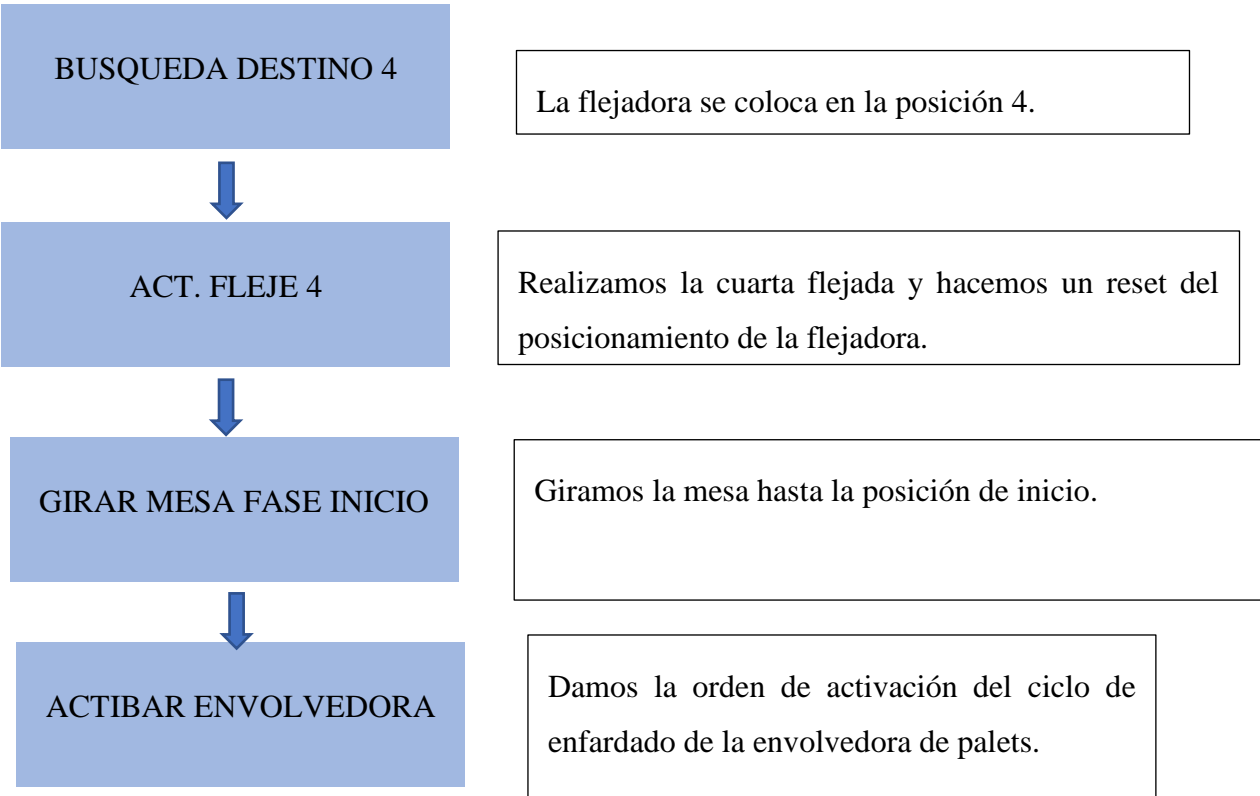


Ilustración 53 Panel para definir programa

4.3 COMUNICACIONES MODEM 2G GPRS GSM

Con la ayuda de la empresa Reisopack, se ha podido implementar el servicio de envío y gestión de datos. Usando su propio servidor y reajustando los parámetros de la plataforma web, podemos acceder a los datos de nuestra máquina desde cualquier ordenar. Únicamente necesitado saber el usuario y la contraseña.

El modem utilizado es el MTX-65i, un modem con tarjeta SIM que la empresa Reisopack lleva utilizando desde hace tiempo, por tanto, nosotros también lo utilizaremos. Podemos ver el modem en la ilustración 54.

MTX-65i

Si buscamos en la Web del fabricante (MTX, 2018), podemos encontrar un poco de información de este aparato.

El MTX-65i es una solución integral que consta de un modem programable en Java J2ME que permite el uso de llamadas GSM, SMS, fax y datos (GPRS clase 12). Gracias a la funcionalidad cuatribanda funciona en todas las frecuencias de GSM. Dispone de una pila TCP/IP de comunicaciones con servicios de Internet que incluyen: TCP, UDP, HTTP, FTP, SMTP, y POP3. Dispone de puertos serie de USB + RS232 + I2C, convertidores analógicos/digitales y GPIOs. Funciona con comandos AT estándar, igual que un modem normal.



Ilustración 54 Modem MTX-65i

4.4 GESTIÓN DE DATOS

La gestión de datos es una de las partes por no decir, la más importante dentro de este proyecto.

La empresa Reisopack, empresa colaboradora en este proyecto y en otros, nos ha enseñado y dado acceso a su servidor para poder realizar el seguimiento de los datos de la WRAPSTER.

Reisopack dispone de un servicio parecido al que pretendemos crear. Al ser una empresa dedicada al fleje, su sistema está centrado en este elemento, pero es fácil mente adaptable a nuestras necesidades.

Gracias al macro que nos instaló Reisopack, para poder extraer los dato del HMI y enviarlos a través del módulo SIM, ahora con acceso a su servidor, podemos hacer las gestiones y control que deseamos:

Página principal:

En la ilustración 55 podemos ver la página principal donde ver la localización de la máquina y podemos ver los últimos estados o valores de los diferentes parámetros.

The screenshot shows the Reisopack CONTROLPACK web interface. The top navigation bar includes 'Component Studio', 'Admin', and user information 'Daniel Souto'. The left sidebar lists components and events. The main content area displays the 'C2200-CONTROLPACK-22171206' component details, including its ID, description, disposition, time zone, creation date, and template. Below this, a table titled 'C2200-CONTROLPACK-22171206: Streams' shows the latest values for various parameters.

Name	Value	Last Updated
(33 Streams)		
alarm_frequency	0.00	2018-04-03 17:13:13 (16h 19m...)
alarm_frequency_2	62.50	2018-04-03 17:13:33 (16h 19m...)
alarm_weighting	0	2018-04-03 17:12:42 (16h 20m...)
alarm_weighting_2	1	2018-04-03 17:12:17 (16h 20m...)
alarmas	code:21	2018-04-03 17:10:15 (16h 22m...)
cellid	214-03-2162-46	2018-04-03 17:12:28 (16h 20m...)

On the right, a map shows the location of the machine with coordinates: Latitude: 40.634507244288585, Longitude: 0.3070659254756285.

Ilustración 55 Página principal del servidor de Reisopack

Gestión de datos:

Como se muestra en la ilustración 56, podemos hacer un seguimiento constante de los datos de producción de la máquina, con su correspondiente gráfica temporal, incluso podemos cruzar otros datos como las provisiones de consumibles que tiene el cliente y poder realizar la orden de aprovisionamiento o incluso indicar el mantenimiento de la máquina.

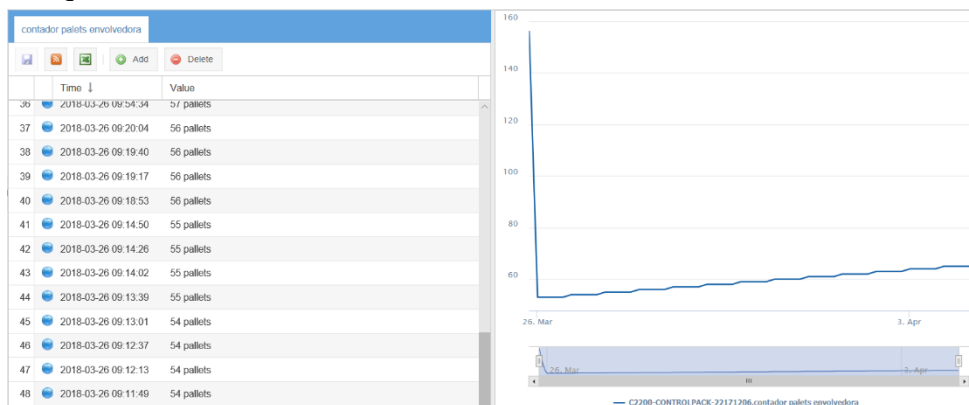


Ilustración 56 Histórico de producción

Alarmas:

alarmas		
	Time ↓	Value
1	2018-04-03 17:10:14	code:21
2	2018-04-03 17:09:54	code:21
3	2018-04-03 17:09:37	code:21
4	2018-04-03 17:09:27	code:21
5	2018-04-03 17:03:05	code:21
6	2018-04-03 17:02:59	code:14
7	2018-03-28 16:50:35	code:21
8	2018-03-28 16:50:04	code:21
9	2018-03-28 16:49:01	code:21
10	2018-03-28 16:46:15	code:21
11	2018-03-28 16:46:13	code:21
12	2018-03-28 16:46:12	code:21
13	2018-03-28 16:46:10	code:21
14	2018-03-28 16:45:55	code:21

En caso de fallo de emergencia o fallo de funcionamiento, se registra dicho error con su correspondiente código como se refleja en la ilustración 57.

Actualmente la única forma de alertar de una avería es mediante las balizas luminosas y acústicas.

Para la siguiente versión de la WRAPSTER se pretende implementar el aviso por llamada de móvil al operario y SMS con la descripción del problema.

Gracias a este sistema también podemos predeterminedir el aprovisionamiento tanto de film como de fleje al cliente. Sabiendo el consumible que se le ha vendido y el consumo,

- ⚡ aprovisionamiento fleje
- 🔥 stock de seguridad fleje
- ⚡ aprovisionamiento cantoneras
- 🔥 stock de seguridad cantoneras

podemos anticiparnos a sus demandas para poderlo reabastecer. La ilustración 58 refleja esta posibilidad donde podemos indicar un mínimo para para que lance el aviso.

Ilustración 58 Histórico de alarmas de la máquina

4.5 TRABAJOS PENDIENTES

4.5.1 PLATAFORMA SMS

La plataforma de control y registro se basará en el sistema PLC S7-1200 de Siemens. Este sistema está integrado en las máquinas Macrobot, pero puede funcionar perfectamente en envolvedoras convencionales.

El sistema está compuesto por las estaciones remotas (PLC's). Para comunicar la estación local con las diferentes estaciones remotas utilizamos tecnología GSM, de transmisión de datos sin cables. Todas las estaciones remotas disponen de módem GSM para tal fin. La información se reporta por SMS a teléfonos convencionales.

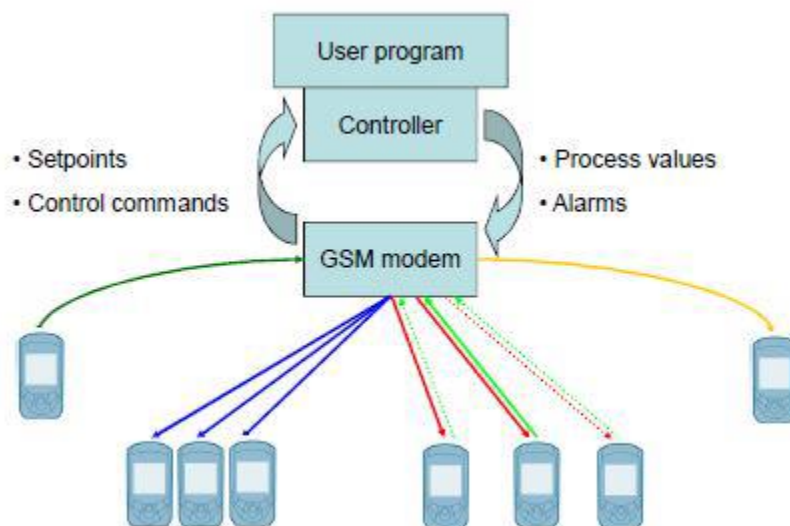


Ilustración 59 diagrama del sistema

La principal función del sistema consiste en registrar los datos de producción a pie de máquina, se registra de manera local al PLC de cada estación remota. La información se va acumulando en las variables de producción:

ACUMULADORES:

- n_ciclos_total
- n_vueltas_total
- n_metros_total

DEFINICIONES INICIALES:

- Matricula de maquina
- Cliente
- Formato (número de vuelta para considerar ciclo correcto)
- Fecha de puesta en marcha
- Habilitación software para envío de SMS
- Números de teléfono para envío de SMS (máximo 9)

Después de inicializar el sistema, la estación enviará por SMS, a los números de teléfono que hayamos dado de alta, la información de producción registrada en la máquina en ese momento.

SMS de producción:

CUSTOMER:	<i>“Cliente ejemplo”</i>
MACHINE NUMBER:	<i>“SPK.6200.xxx”</i>
REGISTER DATE:	<i>“Fecha del controlador”</i>
METERS:	<i>“Total de metros”</i>
LAPS:	<i>“Total de vueltas”</i>
PALETS:	<i>“Total de ciclos”</i>

Ilustración 60 Ejemplo SMS

PETICIÓN DE DATOS DE PRODUCCIÓN: #G#n° telf##PRODUCTION?

(Petición puntual de datos de producción por SMS, se confirma evento en n° telf)

ORDEN ENABLE: #S#n° telf##MACROon=ON

(Habilitar las estaciones remotas por SMS, se confirma evento en n° telf)

ORDEN DISABLE: #S#n° telf##MACROon=OFF

(Deshabilitar las estaciones remotas por SMS, se confirma evento en n° telf)

Elementos:

- PLC
- Módem
- Antena
- Sensores: metros, vueltas y ciclos (entradas digitales)

Requerimientos adicionales:

Para que el módem funcione correctamente debemos insertar una tarjeta SIM en cada estación remota. (Para insertar la SIM el modem debe estar apagado)

Se aconseja utilizar SIM prepago, ya que el gasto de consumo por envío de SMS es bajo. En este caso hay que vigilar la fecha de caducidad de la SIM y realizar recargas con el mínimo importe.

NOTA IMPORTANTE, Si no disponemos de cobertura GSM con la compañía telefónica el sistema

puede llegar a conectarse y no funciona.

4.5.2 CARRO PORTABOBINAS

Tanto la célula de carga como el lector de los tags, se instalarán en el carro de las bobinas. Tal y como podemos ver en el layout de la ilustración 61.

La instalación de estos elementos no requiere ajustar el carro, sino que está pensado para poder acoplar este sistema a máquinas ya existentes con el mínimo de complicaciones.

Como podemos ver el tag se encontrará en el mandril de la bobina, a 10/15 cm del lector RFID.

El lector RFID se instalará en el doble fondo de la base del carro. Este doble fondo se utiliza como sistema de seguridad en caso de encontrarse un obstáculo, el fondo tiene un botón que al levantarse el doble fondo se activa y bloquea la máquina.

La célula de carga, debe estar debajo de la bobina, al ser una célula de compresión, se situará entre el saliente del mandril y la base del carro.

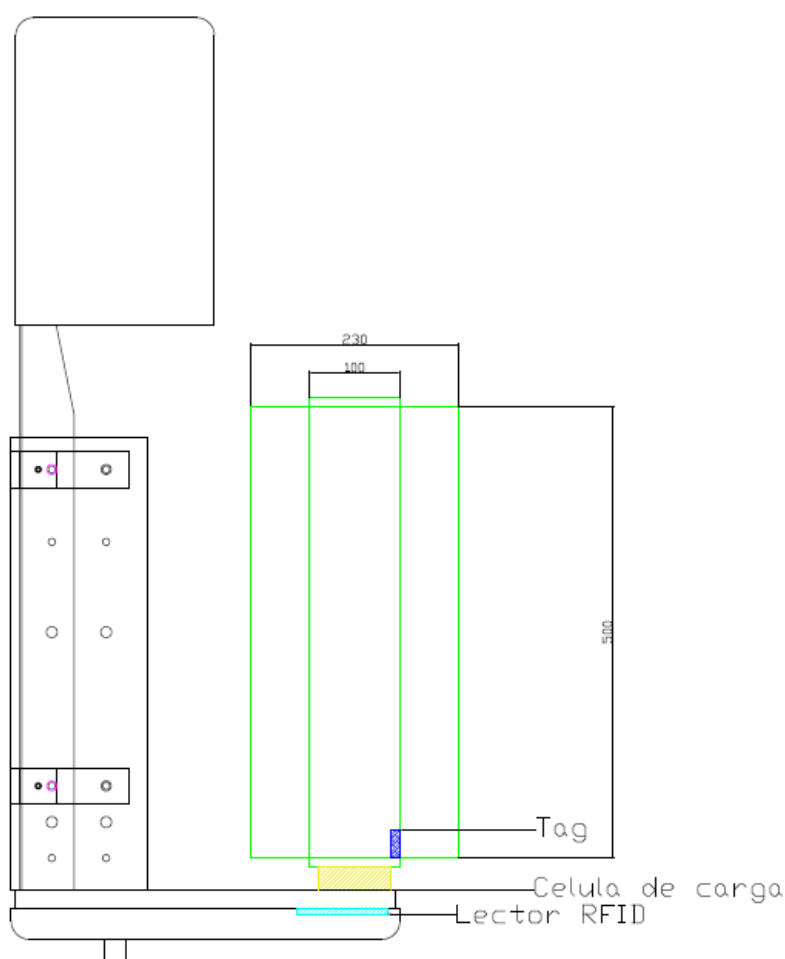


Ilustración 61 Layout carro bobina

El Tag:

Necesitaremos una impresora que de pruebas para poder imprimir las etiquetas con los tags. Los tags magnéticos deberán ser pasivos y almacenar una serie de información.

Al ser una impresora para hacer pruebas debe de ser barata y asequible.



Ilustración 62 Tag de la impresora

Las etiquetas deben tener el tamaño suficiente para ser introducidas de manera manual en el mandril de una bobina sin dificultad.

Como sistema de control, las etiquetan una vez pegadas en el mandril no podrán ser despegadas.

Lector RFID:

Para poder hacer la lectura de los tags, vamos a necesitar un receptor, una antena que pueda leer las etiquetas y enviar esa información al microcontrolador. (label market, 2018).

El sensor que deberemos utilizar y que se tendrá que instalar en nuestras máquinas de enfardado, será el que nos permita hacer una fácil instalación y pueda leer la etiqueta a una distancia de entre 10 y 15 cm.

Deberá ser un lector extra plano semejante al de la ilustración 63.



Ilustración 63 Antena RFID STIX



Ilustración 64 Lector RFID portátil WORKABOUT PRO G4

Por otro lado, también necesitaríamos un lector RFID portátil para poder tener un control sobre las bobinas para poder saber qué clase de material estamos utilizando sin tener que introducir la bobina en una máquina, para ello, el mercado nos ofrece una amplia gama de lectores portátiles que nos resuelven este problema, por ejemplo, el WORKABOUT PRO G4 de la ilustración 64.

Célula de carga:

La teoría muchas veces en la práctica no acaba de funcionar como uno desea.

Anterior mente hemos visto como utilizando un formula y conociendo los parámetros de la máquina y de la misma bobina, podíamos obtener el consumo en gramos de film.

A pesar de que nos pueda dar un valor bastante cercano a la realidad, no tiene en cuenta las irregularidades durante el proceso: mal preestiro, deformación a causa de la tensión, remanentes de film (cola de novia). etc.

Por tanto, la mejor solución es instalar una célula de carga en la base de la bobina, para poder pesarla después de cada ciclo de envoltura, de esa manera obtendríamos el consumo real de la máquina.

Las condiciones que debe cumplir la célula de carga son las siguientes:

Debe poder medir un peso máximo de 20 kg con una resolución de 5 gramos.

Debe tener un diámetro de 100mm y un ancho no superior a 50mm.

Con forma de donut o arandela con un pase central para poder pasar un tornillo que sujete la base del carro con el soporte de la bobina.

Se requiere una célula de carga como la de la ilustración 65.



Ilustración 65 Célula de carga

CONCLUSIONES

Una vez finalizado el presente trabajo, es momento de realizar una reflexión relativa a todo el proceso y poder obtener una serie de conclusiones que permitan asentar conocimientos y procedimientos. De este modo, se pueden enunciar las siguientes:

1. Se ha logrado alcanzar el objetivo fundamental del proyecto: diseñar un prototipo que será la base de un sistema de monitorización, automatización y gestión remota. Queda demostrado que el sistema diseñado y propuesto, cumple con los requerimientos planteados en los objetivos.
2. Se ha demostrado con el prototipo, simulaciones, capturas de pantalla y fotografías que el proceso de diseño es correcto y funcional. Esto conlleva una base en el que será el futuro del SPK-Connect.
3. Se ha alcanzado un conocimiento de software y hardware bastante variado, que de no ser por la realización de proyecto no habría sido posible. Por tanto, esta es una de las conclusiones más importantes que se pueden transmitir, ya que implica conocer nuevas herramientas de diseño y prototipado, para obtener cualquier tipo de solución electrónica.
4. Se ha conseguido poner en práctica competencias comunicativa y la redacción de textos técnicos, sin duda componentes muy importantes en el ámbito de la ingeniería.
5. Se ha realizado un trabajo sobre el terreno al colaborar para poder sacar a delante la WRAPSTER y su sistema basado en SPK-Connect. Esta primera versión de la WRAPSTER es el inicio que hará que en un futuro el SPK-Connect sea plenamente funcional.

Aparte de lo anterior, se considera necesario comentar que ha sido un proceso muy enriquecedor y muy entretenido. Esto es así, dado que me ha permitido

aprender y practicar con el entorno Arduino, algo inicialmente desconocido para mí y que presenta una potencia inimaginable. Quizás la posibilidad de poder montar los circuitos, desarrollar el código y realizar las simulaciones es lo que hace que todo este proceso haya sido ameno; pero también hay que decir que es muy exigente todo el proceso de búsqueda de información y aprendizaje. Con todo, considero que realizar este proyecto en concreto ha sido una muy gran experiencia.

A más, tengo que decir que realizar el estudio previo de un proyecto de empresa que puede que en un futuro se realice, otorga una gran motivación a la hora de realizar este proyecto.

Por otro lado, considerando aplicaciones o futuras líneas de trabajo se pueden enumerar las siguientes:

1. A nivel de hardware, acabar de desarrollar las tecnologías que no se han podido implementar en la WRAPSTER como la lectura de tags, o el cálculo de consumo mediante una célula de carga.
2. Desarrollar un servidor propio de Control Pack, ahora que hemos podido observar el sistema de terceros y ver cómo funciona, ahora es el momento de hacer nuestro propio servicio web e incluso crear una APP propia.
3. En cuanto al software, en el prototipo se podrían incluir una serie de números de terminales móviles de forma que no sea un único usuario el que pueda interactuar con el sistema. Incluso, teniendo en cuenta lo expuesto anteriormente, se podría considerar la opción de acceder al sistema desde internet mediante usuarios registrados y claves de acceso.
4. Monitorizar otros parámetros con alto grado de impacto (parámetros de la máquina basados en el R-connect).
5. Crear un banco de pruebas para poder llevar todo lo mencionado anteriormente a cabo. Disponer de un pequeño espacio donde dispongamos de

todas las tecnologías, poder hacer estudios del consumo de materiales, supervisar el funcionamiento de los diferentes dispositivos y verificar la subida de datos a nuestro servidor, incluso enlazar el servidor con el programa que gestiona el envío y venta de materiales.

En resumen, se pueden considerar diversas mejoras que requieren más tiempo y dedicación del disponibles para el presente trabajo. Es más, se considera suficiente todo lo desarrollado en el presente proyecto, porque se han demostrado conocimientos de programación, diseño y simulaciones con software y montaje de prototipos y simulaciones reales. Con todo y como comentario personal, considero que ha sido un proyecto muy ameno y enriquecedor del que estoy bastante satisfecho. Espero seguir trabajando en este proyecto y un día poderlo finalizarlo.

BIBLIOGRAFÍA

- Arduino.** (2018). *arduino*. Obtenido de <https://www.arduino.cc/>
- Arduino.** (2018). *IDE*. Obtenido de <https://www.arduino.cc/en/Guide/Environment>
- CONTROLPACK.** (2018). Obtenido de <https://www.controlpack.com/>
- digitaltrends.** (13 de diciembre de 2017). *Sistema global para las comunicaciones móviles*. Obtenido de https://es.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones_móviles
- El PAIS.** (17 de mayo de 2007). *Internet de las cosas*. Obtenido de https://elpais.com/diario/2007/05/17/ciberpais/1179368665_850215.html
- FUTUROS.** (15 de junio de 2012). Obtenido de <http://futuros.16mb.com/?futuro=rfid>
- IDIA.** (4 de abril de 2016). Obtenido de <http://www.idia.es/iniciativas/industria-4-0/>
- Kurzweil.** (3 de noviembre de 2011). *IBM open-sources 'Internet of Things' protocol*. Obtenido de <http://www.kurzweilai.net/ibm-open-sources-internet-of-things-protocol>
- label market.** (2018). Obtenido de <http://www.labelmarket.es/>
- Manual comandos AT.** (12 de diciembre de 2010). Obtenido de https://www.espruino.com/datasheets/SIM900_AT.pdf
- movilfacil.** (11 de marzo de 2011). *ARQUITECTURA GSM*. Obtenido de <https://movilfacil.wordpress.com/2011/03/11/18/>
- MTX.** (2018). Obtenido de <http://mtxm2m.com/product/java-modems/mtx-65i-2>
- Prometec.** (17 de marzo de 2016). *Los RFID*. Obtenido de <https://www.prometec.net/los-rfid/>
- Prometec.** (6 de febrero de 2018). *Conjunto de comandos AT*. Obtenido de <https://www.prometec.net/comandos-at-gsm-gprs-gps/>
- Prometec.** (26 de febrero de 2018). *MÓDULO GSM/GPRS: LLAMAR Y ENVIAR SMS*. Obtenido de <https://www.prometec.net/gprs-llamar-enviar-sms/>

Reisopack. (6 de abril de 2018). *Controlpack*. Obtenido de <http://smart.reisopack.com/start.html>

ROBOPAC R-Connect. (6 de Abril de 2018). Obtenido de http://www.robopac.com/ES/rconnect_2

ThingSpeak. (2018). Obtenido de <https://thingspeak.com/>

universia. (15 de julio de 2005). *Servicio general de paquetes vía radio*. Obtenido de http://biblioteca.universia.net/html_bura/ficha/params/id/933581.html

Xively. (2018). Obtenido de <https://xively.com/>

Zebra. (2018). *IMPRESORAS INDUSTRIALES RFID ZT600 SERIES*. Obtenido de <https://www.zebra.com/es/es/products/printers/rfid/zt600-series.html>

ANEXO I CONTROLPACK

CARTA DE PRESENTACION



Somos una empresa con más de 30 años de experiencia dedicados exclusivamente a la comercialización de soluciones para la optimización de los procesos de embalaje. Nuestro Departamento Técnico y Comercial dispone de una amplia experiencia en multitud de sectores y aplicaciones, donde hemos desarrollado todo tipo de proyectos y estudios personalizados de ahorro de costes en el final de línea.

En Controlpack contamos con un equipo de especialistas por áreas (envoltura con film estirable, cerrado de cajas, flejado y retractilado, protección y fijación de la carga, identificación y marcaje, embolsado y envasado).

Gracias a la experiencia de estos años y al trabajo conjunto con nuestros clientes, hemos conseguido convertirnos en su partner tecnológico, ayudándoles a dar un alto valor añadido al embalaje de su producto, de acuerdo a las características concretas de su proceso productivo y todo esto con un coste óptimo.

Estamos plenamente comprometidos con el medio ambiente, nuestro esmero por la reducción de costes de embalaje así lo demuestra.

Además, nuestro departamento I+D+i y Calidad trabajan para la incorporación de materiales más ecológicos, sostenibles y con un mínimo impacto medioambiental.

A continuación, podrá encontrar información detallada de los sistemas tecnológicos disponibles. Así mismo, destacamos una selección de los consumibles en stock permanente, para asegurar un servicio rápido para el cliente, eficiente y a la medida de sus necesidades.

Saludos cordiales.

Manuel Viñals Allepuz

Administrador

Fundador de Controlpack en 1986

SERVICIO TECNICO

En 1986 nació la empresa CONTROLPACK con la idea de llenar un hueco en el mercado del envase y embalaje a base de dar un servicio rápido y completo a todos los clientes.

La idea del servicio esmerado y completo, empezó a dar sus frutos entre las diferentes empresas que poco a poco fueron engrosando la cifra de clientes.

Con el tiempo CONTROLPACK ha ido creciendo hasta consolidarse como una de las empresas punteras en el sector, pero siempre siguiendo la misma filosofía del primer día Servicio rápido y completo.

Siguiendo la idea de ofrecer un buen servicio, se creó el departamento técnico especializado dentro del sector de embalaje, ofreciendo al cliente mantenimiento completo y reparación de la maquinaria de las más importantes marcas del mercado.

Ofrecemos mantenimiento integral para todo tipo de enfardadoras, precintadoras de cajas, retráctiladoras por soldadura, flejadoras tanto manuales, semiautomáticas como automáticas, enfajadoras, etc...

Nuestros técnicos están preparados para cualquier tipo de mantenimiento, reparación o programación en todo tipo de maquinaria de embalaje.



Ilustración 66 Delegaciones y servicio técnico ControlPack

MAQUINARIA

Control Pack es distribuidor de la marca “ROBOPAC” una de las mayores empresas dedicadas al Packaging

A parte, Control pack también tiene su propia línea de maquinaria “SPK”.

A continuación, veremos unos ejemplos de las diferentes líneas de maquinaria de la que disponen.

ENVOLVEDORA DE PALETS



Ilustración 67 Robot S6



Technoplat 3000

Dentro de esta sección podrán descubrir la cantidad de soluciones de las que disponemos, en equipos para la estabilización de cargas paletizadas con film estirable.

La extensa gama de enfardadoras de palets que ofrecemos nos permite abarcar desde la producción espontánea y puntual hasta la altura producción continuada. Disponemos tanto de equipos semiautomáticos como automáticos para solucionar la gran mayoría de aplicaciones que se nos presentan.

ENVOLVEDORAS HORIZONTALES

Ilustración 68 Orbital



Spiror

Máquinas enfajadoras y envolvedoras con film estirable en espiral para una gran variedad de productos de distintos tamaños y formas.

Las aplicaciones típicas son: madera, perfiles de metal y plásticos, tubos de plástico y extrusión de aluminio, periódicos, revistas rollos de textil y parquet, persianas.

Facilidad de manejo con diferentes ciclos de envolvimiento disponibles y parámetros ajustables a cada uno de los productos que se deben envolver. Muy fácil de usar: el operario solamente debe posicionar el producto y presionar el pedal con el pie para iniciar el ciclo; enfajado, enganche del film y corte son operaciones llevadas a cabo automáticamente.

FORMADORAS Y PRECINTADORAS DE CAJAS



Ilustración 69 Robotape CF



SUPERBOX

Las precintadoras son máquinas destinadas al cierre de cajas con cinta adhesiva. Dentro de esta familia cabe destacar también la existencia de máquinas para el formado semiautomático y automático de cajas.

Existen múltiples modelos que van desde la sencillez hasta la automatización total, dependiendo siempre de la necesidad de nuestro cliente.

RETRACTILADORAS

Ilustración 70 SPK Campana

Túnel de retactilado

Este grupo de máquinas se destina a la envoltura con film retráctil. El proceso es simple y ágil y permite la protección de productos variados como libros, folletos, cajas, productos alimentarios, entre otros. Disponemos de una amplia gamma de máquinas para el retractorilado en manual, semiautomático y completamente automático.

ANEXO II CODIGO ARDUINO



```

#include <BasicLinearAlgebra.h>
#include <SPI.h>
#include <MFRC522.h>
#include <SoftwareSerial.h>
#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522
MFRC522 mfrc522(SS_PIN, RST_PIN); ///Creamos el objeto para el RC522
SoftwareSerial SIM900(7,8); // Configura el puerto serie para el SIM900
//declaraciones
char incoming_char; //Variable que guarda los caracteres que envia el SIM900
String mensaje = "";
int LED = 2 ;// def LED Recepcion SMS
int Boton=5;//def Botoen ALARMA
int AlarmaLED=4; //def LED ALARMA
int LEDciclo=3;//def LED CICLO
const int BOTONciclo=6;//def Boton CICLO
int val; //def valor de estado
int contador;//def contador de ciclos
bool estadoalarma = false; //def estado de alarma
byte ActualUID[4]; //almacenará el código del Tag leído
byte Bobina1[4]= {0xA5, 0xA7, 0x22, 0xC3} ; //código de bobina 1
byte Bobina2[4]= {0x13, 0x2E, 0x08, 0x85} ; //código de bobina 2
float consumo; //registro del consumo de film
float peso;//registro del peso de la bobina
float vueltas;//vueltas que da el palet
float micras;//micras del film
float densidadLLDPE=0.916;//constante densidad film
float largobobina=50;//constante altura bobina
float preestiro;//variable preestiro
bool fin_bobina=false;//estado de fin de bobina
bool lectura_bobina=false;//estado de lectura del tag
int alarma=0;//inicio estado alarma

```

```

void setup()
{
  inicializaSIM900();// rutina de inicializacion SIM900
  datos_de_entrada();// definir consumo de film
  pinMode(AlarmaLED,OUTPUT);//def LED como salida
  digitalWrite(Alarma,LOW);//def estado alarma inicial
  Serial.begin(19200); //Iniciamos La comunicacion serial
  SPI.begin();          //Iniciamos el Bus SPI
  mfrc522.PCD_Init(); // Iniciamos el MFRC522
  Serial.println(" ");
  pinMode(LED,OUTPUT); //def LED como salida
  pinMode(Boton,INPUT_PULLUP);// def boton
  incoming_char == '0';
  pinMode(LEDciclo,OUTPUT);//def salida
  pinMode(BOTONciclo,INPUT_PULLUP);//def boton
  contador=0;//poner contador a 0
  infoBOBINA();// identificar TAG
}

void loop()
{
  val=digitalRead(BOTONciclo);//leer estado del boton
  if (val==LOW && fin_bobina==false && lectura_bobina==true)// si se inicia un ciclo...
  {
    digitalWrite(LEDciclo,HIGH);//encender LED ciclo
    contador=contador+1;//incrementar contador
    Serial.print("numero de ciclos: ");
    Serial.print(contador);//mostrar num ciclos
    Serial.println(" ");
    DB();// rutina de subir datos a la web
    Serial.print("gramos de film gastados: ");
    Serial.print(contador*consumo); //consumo acumulado
    Serial.println(" ");
    Serial.print("ciclos restantes: ");
    Serial.print((peso/consumo)-contador);// num Ciclos restantes
    Serial.println(" ");
    delay(2000);
    digitalWrite(LEDciclo,LOW);//fin del ciclo

    if(((peso/consumo)-contador)<=2)// en caso de que queden
    {
      // pocos ciclos
      digitalWrite(AlarmaLED,HIGH);
      delay(1000);
      digitalWrite(AlarmaLED,LOW);// LUZ de alerta
      delay(1000);
      digitalWrite(AlarmaLED,HIGH);
      delay(1000);
      digitalWrite(AlarmaLED,LOW);
      delay(1000);
      digitalWrite(AlarmaLED,HIGH);
      delay(1000);
      digitalWrite(AlarmaLED,LOW);
      Alerta_film();// envio de SMS
    }
  }
}

```

```

    if ((peso/consumo)==contador)// film AGOTADO
    {
        digitalWrite(AlarmaLED,HIGH);
        Alarma_film(); //envio SMS
        llamar();//llamada emergencia
        fin_bobina=true;// estado bobina acabado
    }
}

}

if(lectura_bobina==false)
{
    //si no hay bobina...
    bobina_no_encontrada();
}

    if (SIM900.available() > 0)
    {
        incoming_char = SIM900.read(); //Guardamos el carácter del GPRS
        Serial.print(incoming_char); //Mostramos el carácter en el monitor serie
        mensaje = mensaje + incoming_char; // Añadimos el carácter leído al mensaje
    }
    bool valor = digitalRead(Boton); // leemos boton alarma

    if (incoming_char == '?')//SI envía una SMS con ?...
    {
        digitalWrite(LED,HIGH);
        delay(500);
        digitalWrite(LED,LOW);
        delay(500);
        digitalWrite(LED,HIGH);//indica que se ha recibido un sms
        delay(500);
        digitalWrite(LED,LOW);
        delay(500);
        digitalWrite(LED,HIGH);
        delay(500);
        digitalWrite(LED,LOW);

        if(compareArray(ActualUID,Bobina1))
        {
            mensaje_smsBobina1();// manda un sms con el tipo de bobina
        }//subrutina envio tipo de bobina 1
        if(compareArray(ActualUID,Bobina2))
        {
            mensaje_smsBobina2();// manda un sms con el tipo de bobina
        }//subrutina envio tipo de bobina 2
    }

    if ( valor == false )//si hay alarma...
    {
        Alarma();//subrutina ALARMA
    }
}

```

```

void infoBOBINA()//rutina info bobina
{
  if(mfrc522.PICC_IsNewCardPresent())
  {
    //seleccionamos una tarjeta
    if (mfrc522.PICC_ReadCardSerial())
    {
      // Enviamos serialamente su UID
      Serial.print(F("Card UID:"));
      for (byte i = 0; i < mfrc522.uid.size; i++)
      {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        ActualUID[i]=mfrc522.uid.uidByte[i];
      }
      Serial.print(" ");
      //comparamos los UID para determinar si es uno de nuestras bobinas
      if(compareArray(ActualUID,Bobina1)){// bobina1
        Serial.println("Bobina de control Pack");
        Serial.println("Bobina 23 my 16kg prestiro");
        peso=16000;//def parametros bobina1
        micras=23;
        lectura_bobina=true;
        mfrc522.PICC_HaltA();
      }
      if(compareArray(ActualUID,Bobina2)){// bobina2
        Serial.println("Bobina de control Pack");
        Serial.println("Bobina 30 my 16kg prestiro");
        peso=16000;//def parametros bobina2
        micras=30;
        lectura_bobina=true;
        mfrc522.PICC_HaltA();
      }
      // Terminamos la lectura de la tarjeta tarjeta actual
      mfrc522.PICC_HaltA();
    }
  }
  delay(2000);
}

```

```

void inicializaSIM900()//rutia inicio SIM900
{
    SIM900.begin(19200); //Configura velocidad del puerto serie para el SIM900
    Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
    Serial.println("OK");
    delay (1000);
    SIM900.println("AT + CPIN = \"1995\""); //Comando AT para introducir el PIN de la tarjeta
    delay(25000); //Tiempo para que encuentre una RED
    Serial.println("PIN OK");
    SIM900.print("AT+CLIP=1\r"); // Activa la identificación de llamada
    delay(1000);
    SIM900.print("AT+CMGF=1\r"); //Configura el modo texto para enviar o recibir mensajes
    delay(1000);
    SIM900.print("AT+CNMI=2,2,0,0,0\r"); // Saca el contenido del SMS por el puerto serie del GPRS
    delay(1000);
    Serial.println("esperando ordendes...");
}

void mensaje_smsALARMA()//rutina ALARMA
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\"");
    delay(1000); //Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("ALARMA");// Texto del SMS
    delay(100);
    SIM900.println((char)26); //Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
}

```

```

void mensaje_smsBobina1()//rutina envio info bobina
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\"");
    delay(1000);//Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Bobina 23 my 16kg prestiro");// Texto del SMS
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
    sms_ciclo();//rutina envio info ciclos
}

|
void mensaje_smsBobina2()//rutina bobina2
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\"");
    delay(1000);//Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Bobina 30 my 16kg preestiro");// Texto del SMS
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
    sms_ciclo();//rutina envio info ciclos
}

```



```

boolean compareArray(byte array1[],byte array2[])
{
  //rutina comparacion de bobinas
  if(array1[0] != array2[0])return(false);
  if(array1[1] != array2[1])return(false);
  if(array1[2] != array2[2])return(false);
  if(array1[3] != array2[3])return(false);
  return(true);
}

void Alarma()//rutina ALARMA
{
  alarma=1;//dar un valor simbolico a la alarma
  estadoalarma = !estadoalarma;//compara estado alarma
  digitalWrite(AlarmaLED,HIGH);//encender led de alarma(rojo)
  //registrar alarma en la web
  SIM900.println("AT+CIPSHUT");//cerrar posible conexion
  delay(2000);
  SIM900.println("AT+CIPMUX=0");//abrir conexion internet
  delay(2000);
  SIM900.println("AT+CGATT=1");//abrir acceso internet SIM
  delay(2000);
  SIM900.println("AT+CSTT=\"internet\",\"orange\",\"orange\");//APN,username,pass
  delay(2000);
  SIM900.println("AT+CIICR");
  delay(2000);// establecer nuestra direccion IP
  SIM900.println("AT+CIFSR");
  delay(2000);
  SIM900.println("AT+CIPSTART=\"TCP\",\"184.106.153.149\",\"80\");
  delay(2000);// establecer direccion donde queremos ir
  SIM900.println("AT+CIPSEND=50");
  delay(2000);//establecer modo enviar dato
  SIM900.print("GET /update?key=ARAV5UUQK1JTYGH&field2=");
  delay(2000);//direccion y valor a enviar
  SIM900.println(alarma);
  delay(2000);
  SIM900.println("#026");//fin del envio
  delay(2000);
  alarma=0;//volver alarma a cero
  .....

  delay(2000);
  llamar();//hacer llamada

  mensaje_smsALARMA();//SMS alerta
  //rutina envio ALARMA
}

void llamar()//rutina llamada
{
  SIM900.println ("ATD 628565023;") ; // Número de telefono.
  delay (2000);
  SIM900.println();
  delay(3000); // Esperar 30 segundos...
  SIM900.println ("ATH") ; // Colgar
  Serial.println("Llamada finalizada");
  digitalWrite(AlarmaLED,LOW);
}

```

```

void sms_ciclo()//rutina info ciclo
{
    digitalWrite(LED,HIGH);
    num_ciclos();//envío SMS num ciclos
    gram_palet();//envío SMS gam por palet
    ciclos_resta();// envío SMS ciclos restantes
    digitalWrite(LED,LOW);
}
void num_ciclos()//Rutina SMS num ciclos
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\"");
    delay(1000);//Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Numero de ciclos:");// Texto del SMS
    delay(100);
    SIM900.println(contador);// Texto del SMS
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
}

...

void gram_palet()//Rutina SMS gam por palet
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\"");
    delay(1000);//Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Gramos de film por palet:");
    delay(100);
    SIM900.print(consumo);
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
}

void ciclos_resta()//Rutina SMS ciclos restantes
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\"");
    delay(1000);//Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Ciclos restantes:");
    delay(100);
    SIM900.print((peso/consumo)-contador);
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
}

...

```

```

void ciclos_resta()//Rutina SMS ciclos restantes
{
    Serial.println("Enviando SMS...");
    SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
    delay(1000);
    SIM900.println("AT + CMGS = \"628565023\"");
    delay(1000);//Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
    SIM900.println("Ciclos restantes:");
    delay(100);
    SIM900.print((peso/consumo)-contador);
    delay(100);
    SIM900.println((char)26);//Comando de finalizacion ^Z
    delay(100);
    SIM900.println();
    delay(5000); // Esperamos un tiempo para que envíe el SMS
    Serial.println("SMS enviado");
    incoming_char == '0';
}

int leer()//estructura para leer
{
    //el puerto serial
    int valor=0;
    char cadena[24];
    byte contador=0;
    while(Serial.available()==0)
    {

        delay(5);
    }
    while (Serial.available()>0)
    {
        delay(5);
        cadena[contador]=Serial.read();
        contador++;
    }
    valor=atoi(cadena);
    contador=0;
    return valor;
}

void datos_de_entrada()
{//Pedir datos del programa de enfardado
    Serial.println("Vueltas que dar al palet?");
    vueltas=leer();
    Serial.print(" ");
    Serial.print(vueltas);
    Serial.println(" ");
    Serial.println("Preestiro del film?");
    preestiro=leer();
    Serial.print(" ");
    Serial.print(preestiro);
    Serial.println(" ");
}

```

```

void Alerta_film()
{
  // Rutina SMS aviso de fin de bobina
  Serial.println("Enviando SMS...");
  SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
  delay(1000);
  SIM900.println("AT + CMGS = \"628565023\"");
  delay(1000); //Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
  SIM900.println("ALERTA,NUMERO DE CICLOS RESTANTES:");// Texto del SMS
  delay(100);
  SIM900.println((peso/consumo)-contador);// Texto del SMS
  delay(100);
  SIM900.println((char)26);//Comando de finalizacion ^Z
  delay(100);
  SIM900.println();
  delay(5000); // Esperamos un tiempo para que envíe el SMS
  Serial.println("SMS enviado");
  incoming_char == '0';
}

void Alarma_film()
{
  //Rutina SMS film agotado
  Serial.println("Enviando SMS...");
  SIM900.print("AT+CMGF=1\r"); //Comando AT para mandar un SMS
  delay(1000);
  SIM900.println("AT + CMGS = \"628565023\"");
  delay(1000); //Numero al que vamos a enviar el mensaje: Cambiar asteriscos por n° correcto.
  SIM900.println("FILM GASTADO");// Texto del SMS
  delay(100);
  SIM900.println((char)26);//Comando de finalizacion ^Z
  delay(100);
  SIM900.println();
  delay(5000); // Esperamos un tiempo para que envíe el SMS
  Serial.println("SMS enviado");
  incoming_char == '0';
}

void bobina_no_encontrada()
{
  //Rutina que bloquea la máquina, bobina no valida
  Serial.println("BOBINA NO VALIDA");
  digitalWrite(AlarmaLED,HIGH);
  lectura_bobina==true;// bloqueo de la máquina
}

void DB()
{
  //rutina envio de datos a la web
  consumo=(largobobina*4000);
  consumo=(consumo*micras*densidadLLDPE);
  consumo=(consumo/preestiro);
  consumo=(consumo*vuelatas);
  consumo=(consumo/1000);
  //calculo del consumo de film

  inicializar();//Rutina para escribir en la web
  subir_consumo();//Rutina envío de consumo
  inicializar();//Rutina para escribir en la web
  subir_ciclos(); //Rutina envío de ciclos
  Serial.println(consumo);
  Serial.println(micras);
  Serial.println(preestiro);
  Serial.println(vuelatas);
  Serial.println(peso);
  //visualizar datos puerto serial
}

```

```

void inicializar()
{
  //Rutina para escribir en la web
  SIM900.println("AT+CIPSHUT");
  delay(1000); //cerrar posible conexion
  SIM900.println("AT+CIPMUX=0");
  delay(2000); //abrir conexion internet
  SIM900.println("AT+CGATT=1");
  delay(2000); //activar modo GPRS
  SIM900.println("AT+CSTT=\"internet\", \"orange\", \"orange\");
  delay(2000); //APN, username, pass
  SIM900.println("AT+CIICR");
  delay(2000); // establecer nuestra direccion IP
  SIM900.println("AT+CIFSR");
  delay(2000);
  SIM900.println("AT+CIPSTART=\"TCP\", \"184.106.153.149\", \"80\");
  delay(2000); // establecer direccion donde queremos ir
  SIM900.println("AT+CIPSEND=50");
  delay(2000); //establecer modo enviar dato
}

void subir_consumo()
{
  SIM900.print("GET /update?key=ARAV5UUQGK1JTYGH&field1=");
  delay(2000); //direccion donde colocar el valor
  SIM900.println(consumo);
  delay(2000); //dato que queremos enviar
  SIM900.println("#026"); //fin del envio
}

void subir_ciclos()
{
  SIM900.print("GET /update?key=ARAV5UUQGK1JTYGH&field3=");
  delay(2000); //direccion donde colocar el valor
  SIM900.println((peso/consumo)-contador);
  delay(2000); //dato que queremos enviar
  SIM900.println("#026"); //fin del envio
}

```